DISS. ETH NO. 29599

# TOWARDS BETTER IMAGE AND VIDEO RESTORATION

A thesis submitted to
ETH ZURICH

for the degree of
DOCTOR OF SCIENCES
(Dr. sc. ETH Zurich)

presented by
JINGYUN LIANG
Master of Engineering in Control Science and Engineering
National University of Defense Technology
born 14 February 1995
citizen of China

accepted on the recommendation of
Prof. Dr. Luc Van Gool, examiner
Prof. Dr. William T. Freeman, co-examiner
Prof. Dr. Ming-Hsuan Yang, co-examiner
Prof. Dr. Radu Timofte, co-examiner

2023

## ABSTRACT

Image restoration and video restoration are classical problems in computer vision. They aim to restore the original visual signal from the observed corrupted signal. Due to various factors, such as noise, blurring, downsampling and compression, the inevitable loss or corruption of information occurs during imaging and transmission. This leads to the ill-posed nature of restoration problems. Despite the notable advancements in addressing the problem through deep neural networks, there remains room for further improvement in synthetic and real-world scenarios. In this thesis, we present a practical degradation model, an image restoration model and two video restoration models to improve the restoration performance from different aspects.

Firstly, we propose a practical degradation model for image super-resolution. We first analyze the disadvantages of existing methods and design a complex but practical degradation model that consists of randomly shuffled blur, downsampling and noise degradations. In detail, the blurring effect is modeled through two convolutions utilizing isotropic and anisotropic Gaussian kernels. The downsampling is randomly selected from a set of interpolation techniques. To simulate noise, Gaussian noise is added at varying levels, JPEG compression is employed at different quality factors, and processed camera sensor noise is generated using a reverse-forward camera image signal processing (ISP) pipeline model and a RAW image noise model. For higher complexity, we use above degradations for multiple times in a randomly shuffled way. Experiments on synthetic and real-world images demonstrate that the proposed degradation model can significantly improve the practicability of existing methods.

Secondly, we propose a transformer-based model for image restoration. By regarding image pixels as language tokens, we use the attention mechanism to refine pixel features as a weighted sum of its neighbouring features based on their cosine similarity. To improve the efficiency, the image is partitioned into non-overlapped windows, in which attention is conducted within each window independently. We stack multiple attention layers with residual connections to extract deep image features and shift the image for every other layer to enable cross-window

connection. We demonstrate the superiority of the proposed method on three representative restoration tasks: image super-resolution, image denoising and JPEG compression artifact reduction.

Thirdly, we extend the image transformer model to the video domain. Different from single image restoration, video restoration generally requires to utilize temporal information from multiple adjacent but usually misaligned video frames. Therefore, we propose a transformer-based model with parallel frame prediction and long-range temporal dependency modelling abilities for video restoration. The model is composed of multiple scales, each of which consists of two kinds of modules: temporal reciprocal self attention and parallel warping. The former module divides the video into small clips, on which reciprocal attention is applied for joint motion estimation, feature alignment and feature fusion, while self attention is used for feature extraction. To enable cross-clip interactions, the video sequence is shifted for every other layer. In the second module, parallel warping is used to further fuse information from neighboring frames by parallel feature warping. Experimental results on five tasks, including video super-resolution, video deblurring, video denoising, video frame interpolation and space-time video super-resolution, demonstrate that the proposed method outperforms the previous methods by large margins.

Lastly, we improve the video transformer model by integrating the advantages of recurrent design. It processes local neighboring frames in parallel within a globally recurrent framework. Specifically, it divides the video into multiple clips and uses the previously inferred clip feature to estimate the subsequent clip feature. Within each clip, different frame features are jointly updated with implicit feature aggregation. Across different clips, the guided deformable attention is designed for clip-to-clip alignment, which predicts multiple relevant locations from the whole inferred clip and aggregates their features by the attention mechanism. Extensive experiments on video super-resolution, deblurring, and denoising show that the proposed model achieves state-of-the-art performance on benchmark datasets with balanced model size, testing memory and runtime.

All in all, this thesis contributes to various image and video restoration tasks, achieving state-of-the-art performance on benchmark datasets and real-world data.

# ZUSAMMENFASSUNG

Bildrestaurierung und Videorestaurierung sind klassische Probleme im Bereich der Computer Vision. Ihr Ziel ist es, das ursprüngliche visuelle Signal aus dem beobachteten korrupten Signal wiederherzustellen. Aufgrund verschiedener Faktoren wie Rauschen, Unschärfe, Downsampling und Kompression kommt es während der Bildaufnahme und -übertragung zwangsläufig zu Informationsverlust oder -beschädigung. Dies führt zur schlecht gestellten Natur von Restaurationsproblemen. Trotz der beachtlichen Fortschritte bei der Lösung des Problems durch tiefe neuronale Netzwerke besteht weiterhin Raum für Verbesserungen in synthetischen und realen Szenarien. In dieser Arbeit stellen wir ein praktisches Degradationsmodell, ein Bildrestaurierungsmodell und zwei Videorestaurierungsmodelle vor, um die Restaurierungsleistung aus verschiedenen Gesichtspunkten zu verbessern.

Erstens schlagen wir ein praktisches Degradationsmodell für die Bild-Superauflösung vor. Zunächst analysieren wir die Nachteile bestehender Methoden und entwerfen ein komplexes, aber praktisches Degradationsmodell, das aus zufällig verteilten Unschärfe-, Downsampling- und Rausch-Degradierungen besteht. Im Detail wird der Unschärfeeffekt durch zwei Faltungen mit isotropen und anisotropen Gaußschen Kernen modelliert. Das Downsampling wird zufällig aus einer Reihe von Interpolationstechniken ausgewählt. Zur Simulation von Rauschen wird Gaußsches Rauschen auf unterschiedlichen Pegeln hinzugefügt, JPEG-Kompression wird bei verschiedenen Qualitätsfaktoren verwendet, und verarbeitetes Kamerarauschen wird mithilfe eines rückwärts-vorwärts-Kamerabild-Signalverarbeitungs-ISP-Modells und eines RAW-Bildrauschmodells erzeugt. Für höhere Komplexität verwenden wir die oben genannten Degradierungen mehrmals auf zufällige Weise. Experimente an synthetischen und realen Bildern zeigen, dass das vorgeschlagene Degradationsmodell die Praktikabilität bestehender Methoden signifikant verbessern kann.

Zweitens schlagen wir ein transformerbasiertes Modell für die Bildrestaurierung vor. Indem wir Bildpixel als Sprachtokens betrachten, verwenden wir den Aufmerksamkeitsmechanismus, um die Pixelmerkmale als gewichtete Summe ihrer benachbarten Merkmale aufgrund ihrer Ko-

sinusähnlichkeit zu verfeinern. Zur Verbesserung der Effizienz wird das Bild in nicht überlappende Fenster unterteilt, in denen die Aufmerksamkeit innerhalb jedes Fensters unabhängig durchgeführt wird. Wir stapeln mehrere Aufmerksamkeitsschichten mit Restverbindungen, um tiefe Bildmerkmale zu extrahieren, und verschieben das Bild für jede andere Schicht, um eine Verbindung zwischen den Fenstern zu ermöglichen. Wir zeigen die Überlegenheit der vorgeschlagenen Methode bei drei repräsentativen Restaurierungsaufgaben: Bild-Superauflösung, Bildentrauschen und Reduzierung von JPEG-Kompressionsartefakten.

Drittens erweitern wir das Bildtransformatormodell auf den Videobereich. Anders als bei der Restaurierung einzelner Bilder erfordert die Videorestaurierung im Allgemeinen die Nutzung zeitlicher Informationen aus mehreren benachbarten, aber in der Regel nicht ausgerichteten Videoframes. Daher schlagen wir ein transformerbasiertes Modell mit paralleler Rahmenvorhersage und Fähigkeiten zur Modellierung langreichweitiger zeitlicher Abhängigkeiten für die Videorestaurierung vor. Das Modell besteht aus mehreren Skalen, von denen jede zwei Arten von Modulen enthält: zeitliche reziproke Selbstaufmerksamkeit und paralleles Warping. Das erste Modul teilt das Video in kleine Clips auf, auf denen reziproke Aufmerksamkeit für gemeinsame Bewegungsschätzung, Merkmalsausrichtung und Merkmalsfusion angewendet wird, während Selbstaufmerksamkeit für Merkmalsextraktion verwendet wird. Um Wechselwirkungen zwischen den Clips zu ermöglichen, wird die Videosequenz für jede andere Schicht verschoben. Im zweiten Modul wird paralleles Warping verwendet, um Informationen von benachbarten Frames durch paralleles Merkmalswarping weiter zu fusionieren. Experimentelle Ergebnisse in fünf Aufgaben, einschließlich Video-Superauflösung, Video-Entwirbelung, Videoentrauschen, Video-Bildinterpolation und Raum-Zeit-Video-Superauflösung, zeigen, dass die vorgeschlagene Methode die früheren Methoden deutlich übertrifft.

Zuletzt verbessern wir das Videotransformator-Modell, indem wir die Vorteile des rekurrenten Designs integrieren. Es verarbeitet lokale benachbarte Frames innerhalb eines global rekurrenten Rahmens parallel. Speziell teilt es das Video in mehrere Clips auf und verwendet das zuvor inferierte Clip-Feature, um das nachfolgende Clip-Feature zu schätzen. Innerhalb jedes Clips werden verschiedene Rahmenmerkmale gemeinsam mit impliziter Merkmalsaggregation aktualisiert. Zwischen den verschiedenen Clips ist die geführte deformierbare Aufmerksam-

keit für die Clip-zu-Clip-Ausrichtung konzipiert, die mehrere relevante Positionen aus dem gesamten inferierten Clip vorhersagt und ihre Merkmale durch den Aufmerksamkeitsmechanismus aggregiert. Umfangreiche Experimente zur Video-Superauflösung, -Entwirbelung und -Entzerrung zeigen, dass das vorgeschlagene Modell auf Benchmark-Datensätzen mit ausgewogenen Modellgrößen, Testspeicher und Laufzeit eine Spitzenleistung erzielt.

Insgesamt trägt diese Arbeit zu verschiedenen Bild- und Videorestaurierungsaufgaben bei und erzielt Spitzenleistungen auf Benchmark-Datensätzen und realen Daten.

# PUBLICATIONS

The following publications are included in parts or in an extended version in this thesis:

- J. Liang, J. Cao, G. Sun, *et al.*, „SwinIR: Image restoration using swin transformer," in *IEEE Conference on International Conference on Computer Vision Workshops*, 2021, pp. 1833–1844.

- J. Liang, J. Cao, Y. Fan, *et al.*, „Vrt: A video restoration transformer," *arXiv preprint arXiv:2201.12288*, 2022.

- J. Liang, Y. Fan, X. Xiang, *et al.*, „Recurrent video restoration transformer with guided deformable attention," in *Advances in Neural Information Processing Systems*, 2022, pp. 378–393.

- K. Zhang, J. Liang, L. Van Gool, *et al.*, „Designing a practical degradation model for deep blind image super-resolution," in *IEEE Conference on International Conference on Computer Vision*, 2021, pp. 4791–4800.

Furthermore, the following publications were part of my PhD research, are however not covered in this thesis. The topics of these publications are outside of the scope of the material covered here:

- J. Liang, K. Zhang, S. Gu, *et al.*, „Flow-based kernel prior with application to blind super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 601–10 610.

- J. Liang, G. Sun, K. Zhang, *et al.*, „Mutual affine network for spatially variant kernel estimation in blind image super-resolution," in *IEEE Conference on International Conference on Computer Vision*, 2021, pp. 4096–4105.

- J. Liang, A. Lugmayr, K. Zhang, *et al.*, „Hierarchical conditional flow: A unified framework for image super-resolution and image rescaling," in *IEEE Conference on International Conference on Computer Vision*, 2021, pp. 4076–4085.

- J. Cao, Y. Li, K. Zhang, *et al.*, „Video super-resolution transformer," *arXiv preprint arXiv:2106.06847*, 2021.
- J. Cao, J. Liang, K. Zhang, *et al.*, „Reference-based image super-resolution with deformable attention transformer," in *European Conference on Computer Vision*, 2022, pp. 325–342.
- J. Cao, J. Liang, K. Zhang, *et al.*, „Towards interpretable video super-resolution via alternating optimization," in *European Conference on Computer Vision*, 2022, pp. 393–411.
- L. Sun, C. Sakaridis, J. Liang, *et al.*, „Event-based fusion for motion deblurring with cross-modal attention," in *European Conference on Computer Vision*, 2022, pp. 412–428.
- K. Zhang, Y. Li, J. Liang, *et al.*, „Practical blind denoising via swin-conv-unet and data synthesis," *Machine Intelligence Research*, 2022.
- J. Cao, Q. Wang, J. Liang, *et al.*, „Practical real video denoising with realistic degradation model," *arXiv preprint arXiv:2208.11803*, 2022.
- L. Sun, C. Sakaridis, J. Liang, *et al.*, „Event-based frame interpolation with ad-hoc deblurring," in *Computer Vision and Pattern Recognition*, 2022, pp. 1146–1155.
- Y. Li, K. Zhang, J. Liang, *et al.*, „LSDIR: A large scale dataset for image restoration," in *Computer Vision and Pattern Recognition Workshops*, 2023, pp. 72–81.
- Y. Zhu, K. Zhang, J. Liang, *et al.*, „Denoising diffusion models for plug-and-play image restoration," *arXiv preprint arXiv:2305.08995*, 2023.

# ACKNOWLEDGMENTS

I would like to express my sincere gratitude and appreciation to the following individuals who have played a significant role in the completion of my PhD thesis:

First and foremost, I am deeply indebted to my supervisor, Prof. Dr. Luc Van Gool, for his unwavering support and guidance throughout the past four years. His relentless motivation and encouragement have inspired me to explore various captivating topics in the realm of computer vision. I am immensely grateful for his invaluable ideas and the freedom he granted me to pursue and develop them. It has been an honor to work under his supervision as a member of the Computer Vision Lab.

I would also like to thank my co-supervisor, Prof. Dr. Radu Timofte. His instrumental role in initiating my research on vision restoration and guiding me into the broader field of low-level vision has been invaluable. His early-stage advice served as a beacon, guiding me to find my own path in research and shaping my future career trajectory. The weekly meetings and discussions I shared with him have been instrumental in my academic growth, and I am grateful for the knowledge and insights he imparted.

I also feel deeply grateful to Prof. Dr. William T. Freeman and Prof. Dr. Ming-Hsuan Yang for accepting the invitation to be my co-examiners. Their willingness to read my thesis and thoughtful feedback is a testament to their expertise and generosity. I am honored to have them as esteemed members of my PhD defense committee.

Special appreciation goes to Dr. Kai Zhang, with whom I had the privilege of working closely. From the inception of my research on image super-resolution, he guided me step by step, from brainstorming and model design to conducting experiments and writing papers. His wealth of experience and expertise in vision restoration prevented me from straying off course and taught me invaluable lessons along the way. I am truly grateful for his guidance and the fruitful collaboration we shared throughout my PhD journey.

Within the Computer Vision Lab, I had the pleasure of collaborating with many diligent and brilliant individuals. I would like to thank

my co-authors, Jiezhang Cao, Lei Sun, Shuhang Gu, Martin Danelljan, Andreas Lugmayr, Yawei Li, Christos Sakaridis, Yulun Zhang, Guolei Sun, Wenguan Wang, Qin Wang, Hao Tang, Yun Liu, Ce Liu, Yuanzhi Zhu and others. Working with them on various projects has been an enlightening experience, and I have learned a great deal from our collective efforts. Together, we faced challenges, solved problems, and raced against deadlines, creating countless memorable moments. I would also like to express my gratitude to the other lab members, including Anfeng, Ajad, Cadu, Christ, Christina, Christine, Chunmei, Danda, Dengping, Dengxing, Goutam, Gurkirt, Hanqing, Martin, Mengya, Mohamad, Nico, Prune, Ren, Rui L, Rui G, Suryansh, Thomas, Wen, Xia, Xiaoran, Yuhua, Zhiwu and others. They have been more than just colleagues; they have become my friends. I appreciate the wonderful times we shared over coffee breaks and lunches.

I would like to acknowledge the invaluable mentorship and supervision provided by Dr. Yuchen Fan and Dr. Federico Perazzi from the industrial world. Their guidance has been instrumental in shaping my understanding of the business environment and the requirements of industrial products. Working with them has broadened my horizons and cultivated a multidimensional perspective in my thinking. I would also like to express my gratitude to Rakesh Ranjan, Dr. Xiaoyu Xiang, Prof. Dr. Eddy Ilg, Ryan Wong and Dr. Kaiwen Guo for their contributions to my growth and development.

Finally, I want to dedicate my deepest love and heartfelt gratitude to my family and my girlfriend Yijue. Their unwavering support and boundless love have been my rock throughout these years, especially during the challenging three-year pandemic. They have provided me with a nurturing and comforting harbor amidst the stormy seas of my PhD journey and will always wait for me at home.

# CONTENTS

# LIST OF FIGURES

## LIST OF TABLES

# 1

# INTRODUCTION

In this digital age, a massive amount of images are created and shared every day by smartphones and social media platforms. The process of image creation and sharing encompasses various factors that can potentially impair the quality of the generated images. For example, noises are unavoidable during camera sensing and quantization processes, while motion blurs are common due to object movement or camera shake. In order to enhance visual effects for human or facilitate accurate analysis in visual understanding, the removal of noise and blurring artifacts becomes imperative in certain scenarios. In the field of computer vision, this kind of process is called image restoration. More formally, image restoration refers to the process of restoring clear, high-quality images from degraded, low-quality images. According to the difference of degradations, it can be further divided into several sub-tasks: image super-resolution [1], [17], [18], image denoising [19], [20], image deblurring [21], [22], compression artifact reduction [23], *etc*. For example, image super-resolution aims at reconstructing the high-resolution image from the downsampled low-resolution image, while image denoising aims to remove the noises on images and generate clean sharp images. When the input and output are sequences of video frames, we can define it as the video restoration problem, including video super-resolution [2], [24], video denoising [25], [26], video deblurring [27], [28], video frame interpolation [29], [30], *etc*. In this thesis, image and video restoration are collectively referred to as visual restoration.

In this chapter, we will first introduce the general imaging process and analyze where the degradations come from. Then, we discuss the traditional degradation model and the traditional methods. After that, we discuss the challenges and practical applications of image and video restorations.

## 1.1 THE GENERAL IMAGING PROCESS

In modern digital cameras, the imaging process involves several key steps that work together to capture and produce images. Due to imperfect hardware or inevitable approximation, each step might suffer from different kinds of degradations that reduce image quality. These steps are detailed as follows.

1. Light capture. The lens system is used to capture the reflected or emitted light from the surface. It takes the incoming light rays and focuses them onto the image sensor. In the popular pinhole model, the lens are often assumed to have spherical surfaces and very thin thickness, but the manufactured real lens might not be ideal, leading to geometrical and chromatic aberrations [31]. The former one includes spherical aberration, astigmatism, coma, *etc*, which are visible as image distortions or degradation like blurring. The later one means rays of different wavelengths are focused on different planes due to different refractive indexes. This may cause repeated and shifted structures of different colors.

2. Image sensing. To convert the light to an electrical signal, digital cameras use a regular array of light-sensitive cells to collect incoming photons and turn them into an electric charge by the photoelectric effect. In popular CMOS (complementary metal-oxide semiconductor) cameras, each photo sensor cell has its own transistors that help perform the charge-to-voltage conversion, signal amplification and readout. Due to inherent imperfections in sensors, sensor noises are inevitable. For example, the number of photons captured by the sensor might fluctuate randomly, especially in low-light conditions or under changing temperature conditions. Moreover, the noises might be specific to individual sensor cells, as a result of non-uniform sensor sensitivity, manufacturing defects, signal amplification inconsistencies, *etc*. In addition, owing to the limited intensity of censor cells, the resolution of the camera is also limited. This means that the resulting image is a sampled (downsampled) signal of the real scene.

3. Analog-to-digital conversion. The initial electrical signal captured by the sensor exists in an analog form. To facilitate subsequent processing stages, the signal undergoes analog-to-digital con-

version, wherein it is transformed into a digital format. This conversion entails discretizing the continuous analog signal into discrete values, inevitably leading to a loss of precision.

4. Image signal processing. After light-to-electrical signal conversion, the image signal processing (ISP) algorithm is applied for the enhancement and optimization of the captured raw image. It includes demosaicing, noise reduction, white balance, exposure control, sharpness enhancement, color correction, tone mapping, dynamic range adjustment, *etc*. However, in certain scenarios, the ISP might lead to various degradations or artifacts, such as noise amplification, color distortion, oversharpening and detail loss. This can significantly impact the overall image quality and user experience.

5. Image compression. To save storage and transmission cost, the image data is often compressed to reduce file size. For higher compression rate, the compression process is often lossy, such as JPEG compression. Although it preserves most of the information, some details are inevitably lost and the images may suffer from compression artifacts.

## 1.2 TRADITIONAL DEGRADATION MODEL

Although the imaging process is complicated, it is often simplified and abstracted as various sub-problems in visual restoration research. Based on their distinct degradation assumptions, they can be divided as image/video super-resolution, deblurring, denoising, compression artifact reduction, frame interpolation, *etc*. These sub-problems could be defined by the traditional degradation model.

We begin our definition with video restoration, considering that image restoration can be viewed as a specific instance of video restoration in which both the number of input and output images are one. In general, as shown in Fig. 1.1, a sequence of video frames is captured by a camera with a periodically on-and-off shutter [32], [33]. When the shutter is open, the camera sensors collect reflected photons and convert them into electrical signals. This can be formulated as an integration of luminous intensity over the exposure time, during which the motion blur may occur if the object moves or the camera shakes. Besides, due to limited shutter on-and-off frequency (framerate), motion aliasing

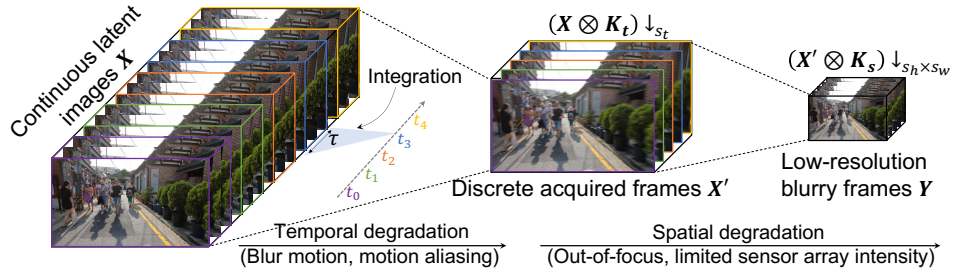



Figure 1.1: The illustration of the traditional degradation model. Note that we show temporal and spatial degradation separately for clarity, although they occur simultaneously. Camera sensors capture discrete frames at the time step $t_i (i \geq 0)$ by integrating the continuous latent images within an exposure time interval $\tau$, leading to temporal degradation. Then, non-ideal imaging factors such as out-of-focus and limited sensor array intensity result in spatial degradation as well. These two degradations can be implemented by using a temporal kernel $\boldsymbol{K}_t$ and a spatial kernel $\boldsymbol{K}_s$ with the downsampling $\downarrow_{s_t}$ and $\downarrow_{s_h \times s_w}$.

may also occur when the temporal dynamic event frequency is beyond the Nyquist limit of framerate. In addition to above temporal degradation, video capturing also suffers from similar spatial degradation to single image capturing as a result of non-ideal imaging factors such as out-of-focus and limited sensor array intensity [6]. Formally, given a high spatio-temporal resolution video $\boldsymbol{X} \in \mathcal{R}^{T_h \times H_h \times W_h \times 3}$, a 3D blur kernel $\boldsymbol{K}$, a low spatio-temporal resolution video $\boldsymbol{Y} \in \mathcal{R}^{T_l \times H_l \times W_l \times 3}$ can be formulated as

$$\boldsymbol{Y} = (\boldsymbol{X} \otimes \boldsymbol{K}) \downarrow_{s_t \times s_h \times s_w} + \boldsymbol{N}, \tag{1.1}$$

where $\otimes$ represents the 3D convolution, and $\downarrow_{s_t \times s_h \times s_w}$ (abbreviated as $\downarrow_s$ for clarity) denotes the standard $s$-fold downsampling in three directions: temporal, vertical and horizontal directions. $\boldsymbol{N}$ is often assumed to be the additive white Gaussian noise with a noise level of $\sigma$. In addition, the sizes of $\boldsymbol{X}$ and $\boldsymbol{Y}$ satisfy $T_h = s_t T_l, H_h = s_h H_l$ and $W_h = s_w W_l$.

Equation (1.1) presents a unified model that encompasses both image and video restoration. Various widely recognized image and video

restoration tasks can be regarded as specific instances or special cases within this unified framework. For instance, in video super-resolution, $\boldsymbol{K}$ is often assumed to be a fixed 2D Gaussian blur kernel and $\boldsymbol{N}$ is omitted. In deblurring, one popular assumption to synthesize blurry frames is averaging neighbouring frames, which equals to using an all-one 1D temporal kernel. Image restoration tasks, such as image super-resolution, image deblurring, and image denoising, can be effectively modeled by reducing the dimensionality of the variables from a 3D space to a 2D space.

MAXIMUM A POSTERIORI (MAP) FRAMEWORK In visual restoration, the target is to estimate the high-quality image or video $\boldsymbol{X}$ given the low-quality observation $\boldsymbol{Y}$ defined in Equation (1.1). According to the Maximum A Posteriori (MAP) framework, we solve the problem by minimizing the energy function $E(\boldsymbol{X})$,

$$\widehat{\boldsymbol{X}} = \underset{\boldsymbol{X}}{\operatorname{argmin}}\, E(\boldsymbol{X}) := \frac{1}{2\sigma^2} \underbrace{\|\boldsymbol{Y} - (\boldsymbol{X} \otimes \boldsymbol{K}) \downarrow_s \|^2}_{\text{data fidelity term}} + \lambda \underbrace{\Phi(\boldsymbol{X})}_{\text{prior term}}, \tag{1.2}$$

where $\lambda$ is a trade-off parameter, the data fidelity term is associated with the model likelihood for reconstruction, and the prior term is a regularization which is related to the prior information of the high spatio-temporal resolution video. However, the prior term is often unknown in practice, and thus it is intractable to directly compute an analytical solution to Problem (1.2).

Traditional methods often try to solve this problem by adding hand-crafted priors [34], [35]. Then, it can use optimization-based methods such as Half-Quadratic Splitting (HQS) algorithm [36], [37] to find an approximate optimal solution. In the era of deep learning, most methods try to learn a deep neural network from large amount of data. They have achieved significantly better performance than traditional methods.

## 1.3 CHALLENGES IN IMAGE AND VIDEO RESTORATION

Despite significant advancements in recent years, the restoration problem remains far from being fully resolved. Several challenges continue to impede its development and hinder the achievement of satisfactory solutions.

1. Complex real-world degradations. In Eq. (1.1), the imaging process is simplified as a combination of blurring, downsampling and noising, which might be over-simplistic and does not perform well on real-world images due to training and testing distribution gaps. To avoid this, some methods try to capture paired images with two different cameras by beam-splitting system or post-alignment techniques. Nevertheless, such a way is expensive for large-scale data collection and might not be able to generalize to other types of cameras. To train restoration models, the more practical way is still synthesizing the low-quality counterpart from high-quality data if we have well-designed degradation models. In Chapter 2, we will propose a practical degradation model that takes various complex degradations into consideration for better generalization to real-world cases.

2. Neural network design for learning image/video priors. Due to various factors, information is inevitably lost or corrupted in degradation, making visual restoration an underdetermined problem. This means that many possible solutions can minimize the data fidelity term in Eq. (1.2) and the task of finding a unique solution is highly intractable. To alleviate the ill-posed nature of visual restoration, one possible way is adding extra prior terms. For example, most deep learning-based methods try to learn the prior implicitly with novel architectural design and learned network weights from low-quality and high-quality data pairs. Although recent years have seen great improvements in this field, there remains room for designing better neural networks that can effectively learn the image/video priors for obtaining plausible and visually pleasing results. In Chapter 3, we will propose an image restoration model that allows for content-based interaction and long-range dependency modelling. In Chapters 4, we will extend the similar idea to the video domain and propose a new multi-scale video restoration model with parallel frame prediction and long-range modelling ability. In Chapters 5, we will continue to improve the video restoration model by processing local neighbouring frames in parallel within a globally recurrent framework, which combines the advantages of both parallel models and recurrent models.

3. Frame alignment and information fusion. Compared with image restoration, other two main challenges in video restoration are frame alignment and information fusion. Current alignment methods are based on either optical flow-based warping or deformable convolution. They heavily rely on accurate optical flow estimation or lack direct interaction among relevant locations. In Chapters 4, we will propose a reciprocal attention module with joint motion estimation, feature alignment and feature fusion in local alignment. In Chapters 5, we will propose the guided deformable attention for adaptive feature fusion of relevant locations from one or multiple frames.

4. Model efficiency. Although improving the restoration quality is critical, the model efficiency is also an important aspect, especially for real-world applications. In Chapters 3, we will propose a small version of the image restoration model that achieves state-of-the-art super-resolution performance with limited model size and computation complexity. In Chapters 5, we will improve the model efficiency by introducing the recurrent architecture, while preserving top performance on video restoration.

## 1.4 APPLICATION OF IMAGE AND VIDEO RESTORATION

Since images and videos are often corrupted in many scenarios, visual restoration can be used to reconstruct the high-quality counterparts for better viewing or for later post-processing. It has a wide range of practical applications across various domains. Some example applications are as below.

1. Photography and videography. To enhance the quality and aesthetics of captured images and videos, restoration techniques are often employed to reduce noise, remove blur, correct color and exposure issues, and improve overall visual quality. This is already available on most modern consumer cameras and cellphones.

2. Medical imaging. To assist the diagnostic procedures, it is important to reduce noise, improve contrast and enhance details for medical images, so as to provide healthcare professionals with clearer and more informative visual representations.

3. Remote sensing. In remote sensing, the obtained images are often affected by atmospheric conditions, such as scattering and light absorption. Using restoration techniques can help remove atmospheric effects, and enhance spatial as well as sepctral details.

4. Entertainment and media. In the entertainment industry, restoration techniques are widely used for various purposes. For instance, it could be used to colorize and enhance old photos and films. It is also useful for perception-aware compression, transmission and visualization.

5. Preprocessing for other vision tasks. Most high-level vision tasks such as recognition reply on good-quality image or video inputs. When there are noise or other artifacts, the performance may drop drastically. Therefore, restoration is sometimes used as a preprocessing step for real-world data. By reducing noise, improving clarity, and restoring details, it can assist in better object recognition, facial identification, tracking and scene understanding in security applications or autonomous systems. For example, before recognition and decision making in autonomous driving systems, it is important to remove the fog in foggy weathers or enhance the details in low-light conditions.

# 2

# PRACTICAL DEGRADATION MODEL

It is widely acknowledged that single image restoration methods would not perform well if the assumed degradation model deviates from those in real images. Although several degradation models have taken additional factors into consideration, such as blur, they are still not effective enough to cover the diverse degradations of real images.

To address this issue, this chapter proposes to design a more complex but practical degradation model that consists of randomly shuffled blur, downsampling and noise degradations. Specifically, the blur is approximated by two convolutions with isotropic and anisotropic Gaussian kernels; the downsampling is randomly chosen from nearest, bilinear and bicubic interpolations; the noise is synthesized by adding Gaussian noise with different noise levels, adopting JPEG compression with different quality factors, and generating processed camera sensor noise via reverse-forward camera image signal processing (ISP) pipeline model and RAW image noise model. To verify the effectiveness of the new degradation model, we have trained a deep super-resolver and then applied it to super-resolve both synthetic and real images with diverse degradations. The experimental results demonstrate that the new degradation model can help to significantly improve the practicability of deep super-resolvers, thus providing a powerful alternative solution for real super-resolution applications.

## 2.1 INTRODUCTION

As a representative task in image restoration, image super-resolution (SR) aims to reconstruct the natural and sharp detailed high-resolution (HR) counterpart $\mathbf{x}$ from a low-resolution (LR) image $\mathbf{y}$ [17], [38]. It has recently drawn significant attention due to its high practical value. With the advance of deep neural networks (DNNs), there is a dramatic upsurge of using feed-forward DNNs for fast and effective SR [7], [18], [39]–[42].

Whereas SR methods map a LR image onto a HR counterpart, degradation models define how to map a HR image to a LR one. Two

representative degradation models are bicubic degradation [43] and traditional degradation [44], [45]. The former generates a LR image via bicubic interpolation. The latter can be mathematically modeled by

$$\mathbf{y} = (\mathbf{x} \otimes \mathbf{k}) \downarrow_{\mathbf{s}} + \mathbf{n}. \tag{2.1}$$

It assumes the LR image is obtained by first convolving the HR image with a Gaussian kernel (or point spread function) $\mathbf{k}$ [46] to get a blurry image $\mathbf{x} \otimes \mathbf{k}$, followed by a downsampling operation $\downarrow_{\mathbf{s}}$ with scale factor $\mathbf{s}$ and an addition of white Gaussian noise $\mathbf{n}$ with standard deviation $\sigma$. Specifically, the bicubic degradation can be viewed as a special case of traditional degradation as it can be approximated by setting a proper kernel with zero noise [37], [47]. The degradation model is generally characterized by several factors such as blur kernel and noise level. Depending on whether these factors are known beforehand or not, DNNs-based SR methods can be broadly divided into non-blind methods and blind ones.

Early non-blind SR methods were mainly designed for bicubic degradations [17]. Although significant improvements on the PSNR [39], [48] and perceptual quality [40], [49] have been achieved, such methods usually do not perform well on real images. It is worth noting that this also holds for deep models trained with a generative adversarial loss. The reason is that blur kernels play a vital role for the success of SR methods [46] and a bicubic kernel is too simple. To remedy this, some works use a more complex degradation model which involves a blur kernel and additive white Gaussian noise (AWGN) and a non-blind network that takes the blur kernel and noise level as conditional inputs [47], [50]. Compared to methods based on bicubic degradation, these tend to be more applicable. Yet, they need an accurate estimation of the kernel and the noise level. Otherwise the performance deteriorates seriously [46]. Meanwhile, only a few methods are specially designed for the kernel estimation of SR [47]. As a further step, some blind methods propose to fuse the kernel estimation into the network design [51], [52]. But such methods still fail to produce visually pleasant results for most real images such as JPEG compressed ones. Along another line of blind SR work with unpaired LR/HR training data, the kernel and the noise are first extracted from the LR images and then used to synthesize LR images from the HR images for paired training [53]. Notably, without kernel estimation, the blind model still has a promising performance. On the other hand, it is difficult to collect accurate blur kernels and

noise models from real images. From the above discussion, we draw two conclusions. Firstly, the degradation model is of vital importance to DNNs-based SR methods and a more practical degradation model is worth studying. Secondly, no existing blind SR models are readily applicable to super-resolve real images suffering from different degradation types. Hence, we see two main challenges: the first is to design a more practical SR degradation model for real images, and the second is to learn an effective deep blind model that can work well for most real images. In this chapter, we attempt to solve these two challenges.

For the first challenge, we argue that blur, downsampling and noise are the three key factors that contribute to the degradation of real images. Rather than utilizing Gaussian kernel induced blur, bicubic downsampling, and simple noise models, we propose to expand each of these factors to more practical ones. Specifically, the blur is achieved by two convolutions with an isotropic Gaussian kernel and an anisotropic Gaussian kernel; the downsampling is more general but includes commonly-used downscaling operators such as bilinear and bicubic interpolations; the noise is modeled by AWGN with different noise levels, JPEG compression noise with different quality factors, and processed camera sensor noise by applying reverse-forward camera image signal processing (ISP) pipeline model and RAW image noise model. Furthermore, instead of using the commonly-used blur/downsampling/noise-addition pipeline, we perform randomly shuffled degradations to synthesize LR images. As a result, our new degradation model involves several more adjustable parameters and aims to cover the degradation space of real images.

For the second challenge, we train a deep model based on the new degradation model in an end-to-end supervised manner. Given a HR image, we can synthesize different realistic LR images by setting different parameters for the degradation model. As such, an unlimited number of paired LR/HR training data can be generated for training. Especially noteworthy is that such training data do not suffer from the misalignment issue. By further taking advantage of the powerful expressiveness and advanced training of DNNs, the deep blind model is expected to produce visually pleasant results for real LR images.

The contributions of this chapter are:

1) A practical SR degradation model for real images is designed. It considers more complex degradations for blur, downsampling

and noise and, more importantly, involves a degradation shuffle strategy.

2) With synthetic training data generated using our degradation model, a blind SR model is trained. It performs well on real images under diverse degradations.

3) To the best of our knowledge, this is the first work to adopt a new hand-designed degradation model for general blind image super-resolution.

4) Our work highlights the importance of accurate degradation modeling for practical applications of DNNs-based SR methods.

## 2.2 RELATED WORK

Since this chapter focuses on designing a practical degradation model to train a deep blind DNN model, we will next give a brief overview on related degradation models and deep blind SR methods.

### 2.2.1 *Degradation Models*

As mentioned in the introduction, existing DNNs-based SR methods are generally based on bicubic downsampling [41], [54] and traditional degradations [5], [18], [55]–[57], or some simple variants [50], [58]–[61]. It can be found that existing complex SR degradation models usually consist of a sequence of blur, downsampling and noise addition. For mathematical convenience, the noise is usually assumed to be AWGN which rarely matches the noise distribution of real images. Indeed, the noise could also stem from camera sensor noise and JPEG compression noise which are usually signal-dependent and non-uniform [62]. Regardless of whether the blur is accurately modeled or not, the noise mismatch suffices to cause a performance drop when super-resolvers are applied to real images. In other words, existing degradation models are wanting when it comes to the complexity of real image degradations. Some works do not consider an explicit degradation model [63], [64]. Instead, they use training data to learn the LR-to-HR mapping which only works for the degradations defined by the training images.

### 2.2.2 *Deep Blind Super-Resolution Methods*

Significant achievements resulted from the design and training of deep non-blind SR networks. This said, applying them for blind SR is a non-trivial issue. It should be noted that blind SR methods are mainly deployed for real SR applications. To that end, different research directions have been tried.

The first direction is to initially estimate the degradation parameters for a given LR image, and then apply a non-blind method to obtain the HR result. Bell-Kligler *et al*. [47] propose to estimate the blur kernel via an internal-GAN method before applying the non-blind ZSSR [65] and SRMD [50] methods. Yet, non-blind SR methods are usually sensitive to errors in the blur kernel, producing over-sharp or over-smooth results.

To remedy this, a second direction aims to jointly estimate the blur kernel and the HR image. Gu *et al*. [51] propose an iterative correction scheme to alternately improve the blur kernel and HR result. Cornillere *et al*. [66] propose an optimization procedure for joint blur kernel and HR image estimation by minimizing the error predicted by a trained kernel discriminator. Luo *et al*. [52] propose a deep alternating network that consists of a kernel estimator module and a HR image restorer module. While promising, these methods do not fully take noise into consideration and thus tend to suffer from inaccurate kernel estimation for noisy real images. As a matter of fact, the presence of noise would aggravate the ill-posedness, especially when the noise type is unknown and complex, and the noise level is high.

A third direction is to learn a supervised model with captured real LR/HR pairs. Cai *et al*. [67] and Wei *et al*. [68] separately established a SR dataset with paired LR/HR camera images. Collecting abundant well-aligned training data is cumbersome however, and the learned models are constrained to the LR domain defined by the captured LR images.

Considering the fact that real LR images rarely come with the ground-truth HR, the fourth direction aims at learning with unpaired training data [69]. Yuan *et al*. [63] propose a cycle-in-cycle framework to first map the noisy and blurry LR input to a clean one and then super-resolve the intermediate LR image via a pre-trained model. Lugmayr *et al*. [64] propose to learn a deep degradation mapping by employing a cycle consistency loss and then generate LR/HR pairs for supervised training. Following a similar framework, Ji *et al*. [53] propose to estimate various

blur kernels and extract different noise maps from LR images and then apply the traditional degradation model to synthesize different LR images. Notably, [53] was the winner of the NTIRE 2020 real-world super-resolution challenge [70], which demonstrates the importance of accurate degradation modeling. Although applying this method to training data corrupted by a more complex degradation seems to be straightforward, it would also reduce the accuracy of blur kernel and noise estimation which in turn results in unreliable synthetic LR images.

## 2.3 PRACTICAL DEGRADATION MODEL

Existing restoration methods are mostly trained on ideal degradation settings or specific degradation spaces defined by the low-quality training data. As a result, there is still a mismatch between the assumed degradation model and the real image degradation model. Furthermore, to the best of our knowledge, no existing deep image restoration model can be readily applied for general real image restoration. Therefore, it is worthwhile to design a practical degradation model to train deep restoration models for real applications. Note that, although denoising and deblurring are related to noisy and blurry image super-resolution, most super-resolution methods tackle the blur, noise and super-resolution in a unified rather than a cascaded framework (see, *e.g.*, [37], [44], [46], [50], [53], [58], [59], [63]–[65], [70], [71]). In this section, we focus on real-world image super-resolution degradation model.

### 2.3.1 *Analysis of Traditional Degradation Models*

Before providing our new practical degradation model, it is useful to mention the following facts on traditional degradation models:

1. According to the traditional degradation model, there are three key factors, *i.e.*, blur, downsampling and noise, that affect the degradations of real images.

2. Since both LR and HR images could be noisy and blurry, it is not necessary to adopt the blur/ downsampling/ noise-addition

pipeline as in the traditional degradation model to generate LR images.

3. The blur kernel space of the traditional degradation model should vary across scales, making it in practice tricky to determine for very large scale factors.

4. While the bicubic degradation is rarely suitable for real LR images, it can be used for data augmentation and is indeed a good choice for clean and sharp image super-resolution.

Inspired by the first fact, a direct way to improve the practicability of degradation models is to make the degradation space of the three key factors as large and realistic as possible. Based on the second fact, we then further expand the degradation space by adopting a random shuffle strategy for the three key factors. Like that, a LR image could also be a noisy, downsampled and blurred version of the HR image. To tackle the third fact, one may take advantage of the analytical calculation of the kernel for a large scale factor from a small one. Alternatively, according to the fourth fact, for a large scale factor, one can apply a bicubic (or bilinear) downscaling before the degradation with scale factor 2. Without loss of generality, this section focuses on designing the degradation model for the widely-used scale factors 2 and 4 in image super-resolution.

### 2.3.2 *The Proposed Degradation Model*

In the following, we will detail the degradation model for the following aspects: blur, downsampling, noise, and random shuffle strategy.

#### 2.3.2.1 *Blur*

Blur is a common image degradation. We propose to model the blur from both the HR space and LR space. On the one hand, in the traditional SR degradation model [44], [65], the HR image is first blurred by a convolution with a blur kernel. This HR blur actually aims to prevent aliasing and preserve more spatial information after the subsequent downsampling. On the other hand, the real LR image could be blurry and thus it is a feasible way to model such blur in the LR space. By further considering that Gaussian kernels suffice for the SR task, we

perform two Gaussian blur operations, *i.e.*, $\mathbf{B}_{\text{iso}}$ with isotropic Gaussian kernels and $\mathbf{B}_{\text{aniso}}$ with anisotropic Gaussian kernels [47], [50], [71]. Note that the HR image or LR image could be blurred by two blur operations (see Sec. 2.3.2.4 for more details). By doing so, the degradation space of blur can be greatly expanded.

For the blur kernel setting, the size is uniformly sampled from $\{7 \times 7, 9 \times 9, \cdots, 21 \times 21\}$, the isotropic Gaussian kernel samples the kernel width uniformly from $[0.1, 2.4]$ and $[0.1, 2.8]$ for scale factors 2 and 4, respectively, while the anisotropic Gaussian kernel samples the rotation angle uniformly from $[0, \pi]$ and the length of each axis for scale factors 2 and 4 uniformly from $[0.5, 6]$ and $[0.5, 8]$, respectively. Reflection padding is adopted to ensure the spatial size of the blurred output stays the same. Since the isotropic Gaussian kernel with width 0.1 corresponds to delta (identity) kernel, we can always apply the two blur operations.

#### 2.3.2.2 *Downsampling*

In order to downsample the HR image, perhaps the most direct way is nearest neighbor interpolation. Yet, the resulting LR image will have a misalignment of $0.5\times(\mathbf{s}-1)$ pixels towards the upper-left corner [37]. As remedy, we shift a centered $21 \times 21$ isotropic Gaussian kernel by $0.5\times(\mathbf{s}-1)$ pixels via a 2D linear grid interpolation method [44], and apply it for convolution before the nearest neighbour downsampling. The Gaussian kernel width is randomly chosen from $[0.1, 0.6 \times \mathbf{s}]$. We denote such a downsampling as $\mathbf{D}^{\mathbf{s}}_{\text{nearest}}$. In addition, we also adopt the bicubic and bilinear downsampling methods, denoted by $\mathbf{D}^{\mathbf{s}}_{\text{bilinear}}$ and $\mathbf{D}^{\mathbf{s}}_{\text{bicubic}}$, respectively. Furthermore, a down-up-sampling method $\mathbf{D}^{\mathbf{s}}_{\text{down-up}} (= \mathbf{D}^{\mathbf{s}/\mathbf{a}}_{\text{down}} \mathbf{D}^{\mathbf{a}}_{\text{up}})$ which first downsamples the image with a scale factor $\mathbf{s}/\mathbf{a}$ and then upscales with a scale factor $\mathbf{a}$ is also adopted. Here the interpolation methods are randomly chosen from bilinear and bicubic interpolations, and $\mathbf{a}$ is sampled from $[1/2, \mathbf{s}]$. Clearly, the above four downsampling methods have a blurring step in the HR space, while $\mathbf{D}^{\mathbf{s}}_{\text{down-up}}$ can introduce upscaling-induced blur in the LR space when $\mathbf{a}$ is smaller than 1. We do not include such kinds of blur in Sec. 2.3.2.1 since they are coupled in the downsampling process. We uniformly sample these four downsampling to downscale the HR image.

#### 2.3.2.3 *Noise*

Noise is ubiquitous in real images as it can be caused by different sources. Apart from the widely-used Gaussian noise, our new degradation model also considers JPEG compression noise and camera sensor noise. We next detail the three noise types.

GAUSSIAN NOISE $\mathbf{N}_{\mathrm{G}}$. The Gaussian noise assumption is the most conservative choice when there is no information about the noise [72]. To synthesize Gaussian noise, the three-dimensional (3D) zero-mean Gaussian noise model $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ [73] with covariance matrix $\mathbf{\Sigma}$ is adopted. Such noise model has two special cases: when $\mathbf{\Sigma} = \sigma^2\mathbf{I}$, where $\mathbf{I}$ is the identity matrix, it turns into the widely-used channel-independent additive white Gaussian noise (AWGN) model; when $\mathbf{\Sigma} = \sigma^2\mathbf{1}$, where $\mathbf{1}$ is a $3 \times 3$ matrix with all elements equal to one, it turns into the widely-used gray-scale AWGN model. In our new degradation model, we always add Gaussian noise for data synthesis. In particular, the probabilities of applying the general case and two special cases are set to 0.2, 0.4, 0.4, respectively. As for $\sigma$, it is uniformly sampled from $\{1/255, 2/255, \cdots, 25/255\}$.

JPEG COMPRESSION NOISE $\mathbf{N}_{\mathrm{JPEG}}$. JPEG is the most widely-used image compression standard for bandwidth and storage reduction. Yet, it introduces annoying $8 \times 8$ blocking artifacts/noise, especially for the case of high compression. The degree of compression is determined by the quality factor which is an integer in the range $[0, 100]$. The quality factor 0 means lower quality and higher compression, and vice versa. If the quality factor is larger than 90, no obvious artifacts are introduced. In our new degradation model, the JPEG quality factor is uniformly chosen from $[30, 95]$. Since JPEG is the most popular digital image format, we apply two JPEG compression steps with possibilities 0.75 and 1, respectively. In particular, the latter one is used as the final degradation step.

PROCESSED CAMERA SENSOR NOISE $\mathbf{N}_{\mathrm{S}}$. In modern digital cameras, the output image is obtained by passing the raw sensor data through the image signal processing (ISP) pipeline. In practice, if the ISP pipeline does not perform a denoising step, the processed sensor noise would deteriorate the output image by introducing non-Gaussian

noise [62]. To synthesize such kind of noise, we first get the raw image from an RGB image via the reverse ISP pipeline, and then reconstruct the noisy RGB image via the forward pipeline after adding noise to the synthetic raw image. The raw image noise model is borrowed from [74]. According to the Adobe Digital Negative (DNG) Specification [75], our forward ISP pipeline consists of demosaicing, exposure compensation, white balance, camera to XYZ (D50) color space conversion, XYZ (D50) to linear RGB color space conversion, tone mapping and gamma correction. For demosaicing, the method in [76] which is the same as matlab's `demosaic` function, is adopted. For exposure compensation, the global scaling is chosen from $[2^{-0.1}, 2^{0.3}]$. For the white balance, the red gain and blur gain are uniformly chosen from $[1.2, 2.4]$. For camera to XYZ (D50) color space conversion, the $3 \times 3$ color correction matrix is a random weighted combination of `ForwardMatrix1` and `ForwardMatrix2` from the metadata of raw image files. For the tone mapping, we manually select the best fitted tone curve from [77] for each camera based on paired raw image files and the RGB output. We use five digital cameras, including the Canon EOS 5D Mark III and IV cameras, Huawei P20, P30 and Honor V8 cameras, to establish our ISP pipeline pool. Note that the tone curve and forward color correction matrix do not necessarily come from the same camera. Since tone mapping is not reversible and would result in color shift issue, one should apply the reverse-forward tone mapping for the HR image. We apply this noise synthesis step with a probability of 0.25.

#### 2.3.2.4 *Random Shuffle*

Though simple and mathematically convenient, the traditional degradation model can hardly cover the degradation space of real LR images. On the one hand, the real LR image could also be a noisy, blurry, downsampled, and JPEG compressed version of the HR image. On the other hand, the degradation model which assumes the LR image is a bicubicly downsampled, blurry and noisy version of the HR image can also be used for SR [51], [57]. Hence, a LR image can be degraded by blur, downsampling, and noise with different orders. We thus propose a random shuffle strategy for the new degradation model. Specifically, the degradation sequence $\{\mathbf{B}_{\texttt{iso}}, \mathbf{B}_{\texttt{aniso}}, \mathbf{D}^{\mathbf{s}}, \mathbf{N}_{\mathrm{G}}, \mathbf{N}_{\mathrm{JPEG}}, \mathbf{N}_{\mathrm{S}}\}$ is randomly shuffled, here $\mathbf{D}^{\mathbf{s}}$ represents the downsampling operation with scale factor $\mathbf{s}$ which is randomly chosen from $\{\mathbf{D}^{\mathbf{s}}_{\text{nearest}}, \mathbf{D}^{\mathbf{s}}_{\text{bilinear}}, \mathbf{D}^{\mathbf{s}}_{\text{bicubic}}, \mathbf{D}^{\mathbf{s}}_{\text{down-up}}\}$.

In particular, the sequence of $\mathbf{D}^{\mathbf{s}/\mathbf{a}}_{\text{down}}$ and $\mathbf{D}^{\mathbf{a}}_{\text{up}}$ for $\mathbf{D}^{\mathbf{s}}_{\text{down-up}}$ can insert other degradations. Note that a similar idea of random shuffle strategy was proposed in [78], however, it is designed for image classification and object detection and could be instead used to augment HR images.

With the random shuffle strategy, the degradation space can be expanded substantially. Firstly, other degradation models, such as bicubic and traditional degradation models, and the ones proposed in [51], [57], are special cases of ours. Secondly, the blur degradation space is enlarged by different arrangements of the two blur operations and one of the four downsampling methods. Thirdly, the noise characteristics could be changed by the blur and downsampling, thus expanding the degradation space. For example, the downsampling can reduce the noise strength and make the noise (*e.g.*, processed camera sensor noise and JPEG compression noise) less signal-dependent, whereas $\mathbf{D}^{\mathbf{a}}_{\text{up}}$ ($\mathbf{a} < 1$) can make the signal-independent Gaussian noise to be signal-dependent. Such kinds of noise could exist in real images.

Fig. 2.1 illustrates the proposed degradation model. For a HR image, we can generate different LR images with a wide range of degradations by shuffling the degradation operations and setting different degradation parameters.

### 2.3.3 *Discussion*

It is necessary to add discussion to further understand the proposed new degradation model. Firstly, the degradation model is mainly designed to synthesize degraded LR images. Its most direct application is to train a deep blind super-resolver with paired LR/HR images. In particular, the degradation model can be performed on a large dataset of HR images to produce unlimited perfectly aligned training images, which typically do not suffer from the limited data issue of laboriously collected paired data and the misalignment issue of unpaired training data. Secondly, the degradation model tends to be unsuited to model a degraded LR image as it involves too many degradation parameters and also adopts a random shuffle strategy. Thirdly, the degradation model can produce some degradation cases that rarely happen in real-world scenarios, while this can still be expected to improve the generalization ability of the trained deep blind super-resolver. Fourthly, a DNN with large capacity has the ability to handle different degradations via a single model (see, *e.g.*, [19]). It is worth noting that even when the super-

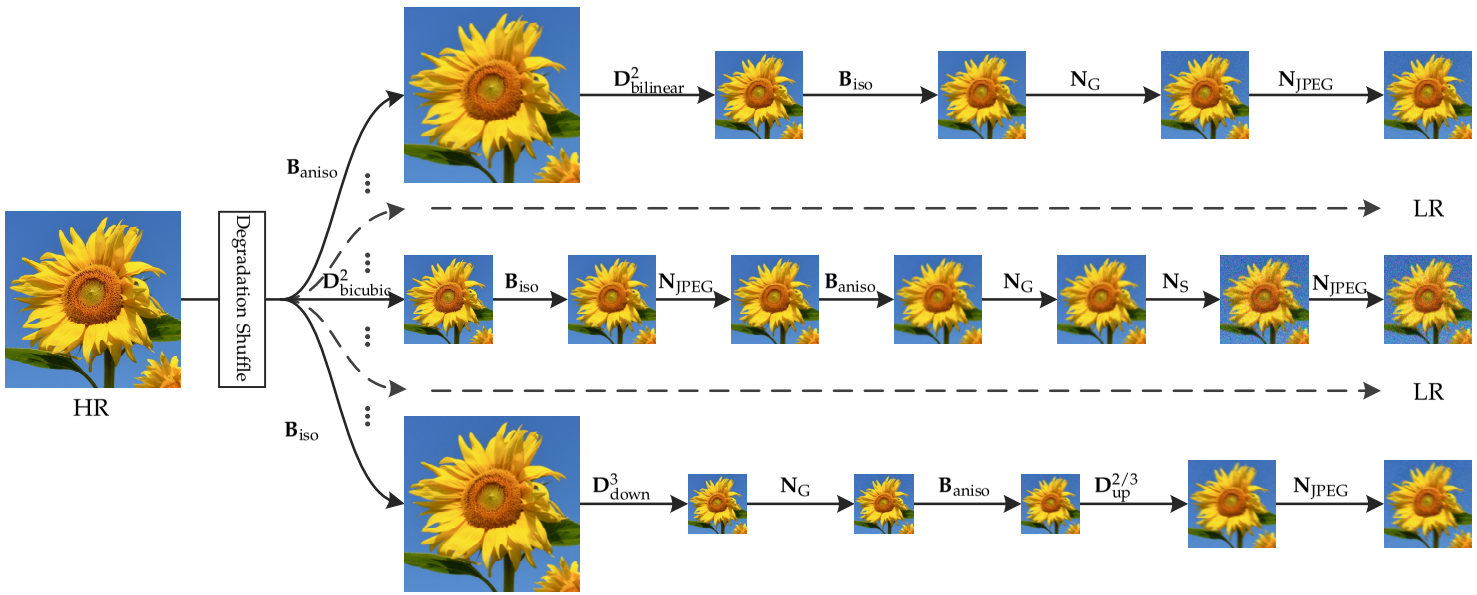



Figure 2.1: Schematic illustration of the proposed degradation model for scale factor 2. For scale factor 4, we additionally apply a bilinear or bicubic downscaling before the degradation for scale factor 2 with a probability of 0.25. For a HR image, the randomly shuffled degradation sequences $\{\mathbf{B}_{\texttt{iso}}, \mathbf{B}_{\texttt{aniso}}, \mathbf{D}^2, \mathbf{N}_{\text{G}}, \mathbf{N}_{\text{JPEG}}, \mathbf{N}_{\text{S}}\}$ are first performed, then a JPEG compression degradation $\mathbf{N}_{\text{JPEG}}$ is applied to save the LR image into JPEG format. The downscaling operation with scale factor 2, *i.e.*, $\mathbf{D}^2$, is uniformly chosen from $\{\mathbf{D}^2_{\text{nearest}}, \mathbf{D}^2_{\text{bilinear}}, \mathbf{D}^2_{\text{bicubic}}, \mathbf{D}^2_{\text{down-up}}\}$.

resolver reduces the performance for unrealistic bicubic downsampling, it is still a preferred choice for real SR. Fifthly, one can conveniently modify the degradation model by changing the degradation parameter settings and adding more reasonable degradation types (*e.g.*, speckle noise and unaligned double JPEG compression [79]) to improve the practicability for certain applications.

## 2.4 EXPERIMENTS

### 2.4.1 *Experimental Setup*

The novelty of this chapter lies in the new degradation model and the possibility of existing network structures such as ESRGAN [40] to be borrowed to train a deep blind model. For the sake of showing the advantage of the proposed degradation model, we adopt the widely-used ESRGAN network and train it with the synthetic LR/HR paired images produced by the new degradation model. Following ESRGAN, we first train a PSNR-oriented BSRNet model and then train the perceptual quality-oriented BSRGAN model. Since the PSNR-oriented BSRNet model tends to produce oversmoothed results due to the pixel-wise average problem [49], the perceptual quality-oriented model is preferred for real applications [80]. Thus, unless otherwise specified, we focus more on the BSRGAN model.

Compared to ESRGAN, BSRGAN is modified in several ways. First, we use a slightly different HR image dataset which includes DIV2K [81], Flick2K [39], [43], WED [82] and 2,000 face images from FFHQ [83] to capture the image prior. The reason is that the goal of BSRGAN is to solve the problem of general-purpose blind image super-resolution, and apart from the degradation prior, an image prior could also contribute to the success of a super-resolver. We also remove the blurry images based on the variance of the Laplacian of an image. Secondly, BSRGAN uses a larger LR patch size of $72 \times 72$. The reason is that our degradation model can produce severely degraded LR images and a larger patch can enable deep models to capture more information for better restoration. Thirdly, we train the BSRGAN by minimizing a weighted combination of L1 loss, VGG perceptual loss and spectral norm-based least square PatchGAN loss [84] with weights 1, 1 and 0.1, respectively. In particular, the VGG perceptual loss is operated on the fourth convolution before the fourth rather than the fifth maxpooling layer of the pre-trained

19-layer VGG model as it is more stable to prevent color shift issues. We train BSRGAN with Adam, using a fixed learning rate of $1 \times 10^{-5}$ and a batch size of 48.

### 2.4.2 *Testing Datasets*

Existing blind SR methods are generally evaluated on specifically designed synthetic data and only very few real images. For example, IKC [51] is evaluated on the blurred, bicubicly downsampled synthetic LR images and two real images; KernelGAN [47] is evaluated on the synthetic DIV2KRK dataset and two real images. As a result, to the best of our knowledge, a real LR image dataset with diverse blur and noise degradations is still lacking.

In order to pave the way for the evaluation of blind SR methods, we establish two datasets, including the synthetic DIV2K4D dataset which contains four subdatasets with a total of 400 images generated from the 100 DIV2K validation images with four different degradation types and the real RealSRSet which consists of 20 real images either downloaded from the internet or directly chosen from existing testing datasets [20], [85]–[87]. Specifically, the four degradation types for DIV2K4D including 1) type I: the commonly-used bicubic degradation; 2) type II: anisotropic Gaussian blur with nearest downsampling by a scale


(a) Examples from DIV2K4D (b) Examples from RealSRSet

Figure 2.2: Some example images from the DIV2K4D and RealSRSet datasets. From top to bottom of (a), we show example images generated by the degradation types II, III and IV.

factor of 4; 3) type III: anisotropic Gaussian blur with nearest downsampling by a scale factor of 2 and subsequent bicubic downsampling by another scale factor of 2 and final JPEG compression with quality factors uniformly sampled from $[41, 90]$; and 4) type IV: our proposed degradation model. Note that the subdataset with degradation type II and the downsampled images by a scale factor of 2 for subdataset with degradation type III are directly borrowed from the DIV2KRK dataset [47]. Some example images from the two datasets are shown in Fig. 2.2, from which we can see the LR images are corrupted by diverse blur and noise degradations. We argue that a general-purpose blind super-resolver should achieve a good overall performance on the two datasets.

### 2.4.3 *Compared Methods*

We compare the proposed BSRNet and BSRGAN with RRDB [40], IKC [51], ESRGAN [40], FSSR-DPED [88], FSSR-JPEG [88], RealSR-DPED [53] and RealSR-JPEG [53]. Specifically, RRDB and ESRGAN are trained on bicubic degradation; IKC is a blind model trained with different isotropic Gaussian kernels; FSSR-DPED and RealSR-DPED are trained to maximize the performance on the blurry and noisy DPED dataset; FSSR-JPEG is trained for JPEG image super-resolution; RealSR-JPEG is a recently released and unpublished model on github. Note that since our novelty lies in the degradation model, and RRDB, ESRGAN, FSSR-DPED, FSSR-JPEG, RealSR-DPED and RealSR-JPEG use the same network architecture as ours, we thus did not re-train other models for comparison.

### 2.4.4 *Experiments on the DIV2K4D Dataset*

The PSNR and LPIPS (learned perceptual image patch similarity) results of different methods on the DIV2K4D datasets are shown in Table 2.1. Note that LPIPS is used to measure the perceptual quality, and a lower LPIPS value means the super-resolved image is more perceptually similar to the ground-truth. We draw several conclusions from Table 2.1. Firstly, as expected, RRDB and ESRGAN perform well for bicubic degradation but do not perform well on non-bicubic degradation as they are trained with the simplified bicubic degradation. It is worth

Table 2.1: The PSNR and LPIPS results of different methods on the DIV2K4D dataset. The PSNR results are calculated on Y channel of YCbCr space.

| Degradation Type | Metric | RRDB | IKC | ESRGAN | FSSR -DPED | FSSR -JPEG | RealSR -DPED | RealSR -JPEG | **BSRNet (ours)** | **BSRGAN (ours)** |
|---|---|---|---|---|---|---|---|---|---|---|
| Type I (Bicubic) | PSNR | 30.89 | 29.95 | 28.16 | 24.55 | 22.71 | 21.72 | 27.35 | 29.07 | 27.30 |
| | LPIPS | 0.254 | 0.263 | 0.115 | 0.240 | 0.364 | 0.312 | 0.213 | 0.331 | 0.236 |
| Type II | PSNR | 25.66 | 27.35 | 25.56 | 25.81 | 25.33 | 26.29 | 25.36 | 27.76 | 26.26 |
| | LPIPS | 0.542 | 0.392 | 0.526 | 0.460 | 0.399 | 0.263 | 0.479 | 0.397 | 0.284 |
| Type III | PSNR | 26.70 | 26.72 | 26.21 | 25.83 | 23.25 | 22.82 | 26.72 | 27.59 | 26.28 |
| | LPIPS | 0.517 | 0.504 | 0.436 | 0.392 | 0.376 | 0.379 | 0.360 | 0.419 | 0.284 |
| Type IV | PSNR | 24.03 | 24.01 | 23.68 | 23.62 | 22.40 | 22.97 | 23.85 | 25.67 | 24.58 |
| | LPIPS | 0.659 | 0.641 | 0.599 | 0.589 | 0.597 | 0.528 | 0.589 | 0.506 | 0.361 |

PSNR↑/LPIPS↓
(a) LR (×4) | 23.51/0.601 (b) IKC [51] | 23.21/0.353 (c) FSSR-JPEG [88] | 23.46/0.504 (d) RealSR-JPEG [53] | 25.48/0.353 (e) **BSRNet (ours)** | 24.65/0.233 (f) **BSRGAN (ours)**

Figure 2.3: Results of different methods on super-resolving a LR image from the DIV2K4D dataset with scale factor 4. The testing image is synthesized by our proposed degradation (*i.e.*, degradation type IV).

noting that, even trained with GAN, ESRGAN can slightly improve the LPIPS values over RRDB on degradation types II-IV. Secondly, FSSR-DPED, FSSR-JPEG, RealSR-DPED and RealSR-JPEG outperform RRDB and ESRGAN in terms of LPIPS since they consider a more practical degradation. Thirdly, for degradation type II, IKC obtains promising PSNR results while RealSR-DPED achieves the best LPIPS result as they are trained on a similar degradation. For degradation types III and IV, they suffer a severe performance drop. Fourthly, our proposed BSRNet achieves the best overall PSNR results, while BSRGAN yields the best overall LPIPS results.

Fig. 2.3 shows the results of different methods on super-resolving a LR image from the DIV2K4D dataset. It can be seen that IKC and RealSR-JPEG fail to remove the noise and to recover sharp edges. On the other hand, FSSR-JPEG can produce sharp images but also introduces some artifacts. In comparison, our BSRNet and BSRGAN produce better visual results than the other methods.

### 2.4.5 *Experiments on the RealSRSet Dataset*

Since the ground-truth for the RealSRSet dataset is not available, we adopt the non-reference image quality assessment (IQA) metrics including NIQE [89], NRQM [90] and PI [91] for quantitative evaluation. As one can see from Table 2.2, BSRGAN fails to show promising results. Yet, as shown in Fig. 2.4, BSRNet produces much better visual results than the other methods. For example, BSRGAN can remove the unknown processed camera sensor noise for "*Building*" and unknown complex noise for "*Oldphoto2*", while also producing sharp edges and fine details. In contrast, FSSR-JPEG, RealSR-DPED and RealSR-JPEG produce some high-frequency artifacts but have better quantitative results than BSRNet. Such inconsistencies indicate that these no-reference IQA metrics do not always match perceptual visual quality [70] and the IQA metric could be updated with new SR methods [92]. We further argue that the IQA metric for SR should also be updated with new image degradation types, which we leave for future work. We note that our BSRGAN tends to produce 'bubble' artifacts in texture region, which may be solved by new loss function or more training data with diverse textures.

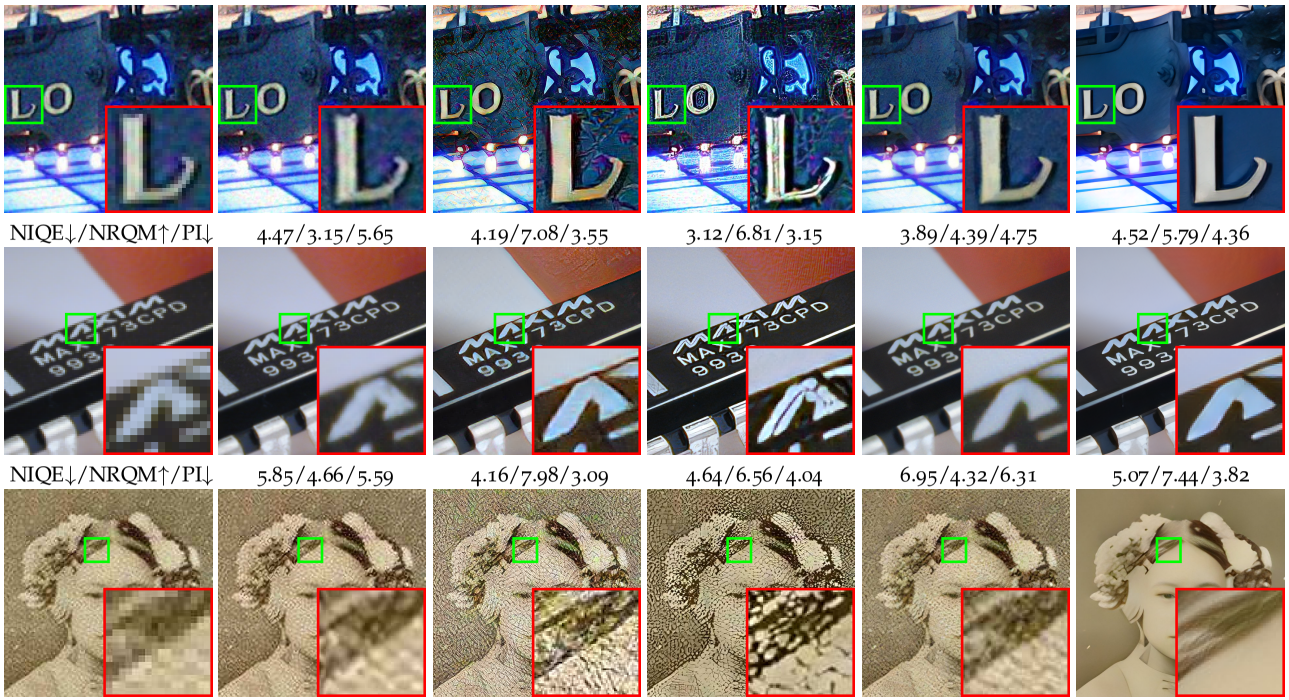



Figure 2.4: Results of different methods on super-resolving real images from RealSRSet with scale factor 4. The LR images from top to bottom in each row are "*Building*", "*Chip*", and "*Oldphoto2*", respectively. Please zoom in for better view.

Table 2.2: The no-reference NIQE [89], NRQM [90] and PI [91] results of different methods on the RealSRSet dataset. The best and second best results are highlighted in red and blue, respectively. Note that all the methods use the same network architecture.

| Metric | ESRGAN | FSSR -DPED | FSSR -JPEG | RealSR -DPED | RealSR -JPEG | **BSRGAN (ours)** |
|---|---|---|---|---|---|---|
| NIQE↓ | 4.95 | 4.86 | 4.04 | 4.58 | 3.99 | 5.60 |
| NRQM↑ | 6.02 | 6.28 | 6.88 | 6.59 | 6.23 | 6.17 |
| PI↓ | 4.47 | 4.29 | 3.58 | 3.99 | 4.29 | 4.72 |

## 2.5 CONCLUSION

In this chapter, we have designed a new degradation model to train a deep blind super-resolution model. Specifically, by making each of the degradation factors, *i.e.*blur, downsampling and noise, more intricate and practical, and also by introducing a random shuffle strategy, the new degradation model can cover a wide range of degradations found in real-world scenarios. Based on the synthetic data generated by the new degradation model, we have trained a deep blind model for general image super-resolution. Experiments on synthetic and real image datasets have shown that the deep blind model performs favorably on images corrupted by diverse degradations. We believe that existing deep super-resolution networks can benefit from our new degradation model to enhance their usefulness in practice. As a result, this work provides a way towards solving blind super-resolution for real applications.

# 3 IMAGE RESTORATION TRANSFORMER

Image restoration refers to the process of improving the quality of an image that has been degraded or corrupted during acquisition, transmission or storage, due to various factors such as noise, object motion, camera motion, quantization and compression. Different sub-tasks, including image super-resolution, image denoising and compression artifact removal, are defined according to different assumptions of degradation models, so as to fit different application scenarios.

This chapter introduces a unified image restoration model for different restoration tasks. It consists of three parts: shallow feature extraction, deep feature extraction and high-quality image reconstruction. In particular, the deep feature extraction module is composed of several residual Swin Transformer blocks, each of which has several Swin Transformer layers together with a residual connection. We conduct experiments on three representative benchmark tasks: image super-resolution (including classical and lightweight image super-resolution), image denoising (including grayscale and color image denoising) and JPEG compression artifact reduction. Experimental results demonstrate that the proposed method outperforms state-of-the-art methods on different tasks by up to 0.14∼0.45dB, while the total number of parameters can be reduced by up to 67%.

## 3.1 INTRODUCTION

Image restoration, such as image super-resolution (SR), image denoising and JPEG compression artifact reduction, aims to reconstruct the high-quality clean image from its low-quality degraded counterpart (*e.g.*, downscaled, noisy and compressed images). Since several revolutionary work [17], [19], [58], [93], convolutional neural networks (CNN) have become the primary workhorse for image restoration [4], [18], [20], [39], [40], [49], [50], [61], [88], [94].

Most CNN-based methods focus on elaborate architecture designs such as residual learning [39], [49] and dense connections [40], [48]. Although the performance is significantly improved compared with

traditional model-based methods [38], [95], [96], they generally suffer from two basic problems that stem from the basic convolution layer. First, the interactions between images and convolution kernels are content-independent. Using the same convolution kernel to restore different image regions may not be the best choice. Second, under the principle of local processing, convolution is not effective for long-range dependency modelling. Although this problem could be alleviated by increasing the network depth and width [97], [98], the number of model parameters is also significantly improved.

As an alternative to CNN, Transformer [99] designs a self-attention mechanism to capture global interactions between contexts and has shown promising performance in several high-level vision problems [100]–[103]. However, vision Transformers for image restoration [8], [104] usually divide the input image into patches with fixed size (*e.g.*, $48\times48$) and process each patch independently. Such a strategy inevitably gives rise to two drawbacks. First, it neglects pixel-wise (*i.e.*, intra-patch) interactions, as it fuses the image content within a patch before attention. This would pose an extra burden on restoring pixels from a patch. The intra-patch information is important in image restoration since it contains original spatial structure of patch. In this sense, it may have poor performance for image restoration which undergo local or pixel-wise degradation. A more natural choice for image restoration is to use pixel-wise attention, which operates in a local patch of the image and computes interactions between pixels. There are some attempts [105], [106] in this direction, but the significant memory and computation requirements pose an obstacle for high-resolution images. Second, the introduced global interactions (*i.e.*, patch-wise attention) might not be suitable for the restoration task, which often has local contents and local degradations. For example, the reconstruction of trees and bricks generally have no relation to each other when they appear in different positions of a single image. Third, the testing image size is often required to be fixed. In this case, border pixels cannot utilize neighbouring pixels that are out of the patch for image restoration. The restored image may introduce border artifacts around each patch. While this problem can be alleviated by patch overlapping, it would introduce extra computational burden.

Recently, Swin Transformer [103] has shown great promise as it integrates the advantages of both CNN and Transformer. On the one hand, it has the advantage of CNN to process image with large size due to

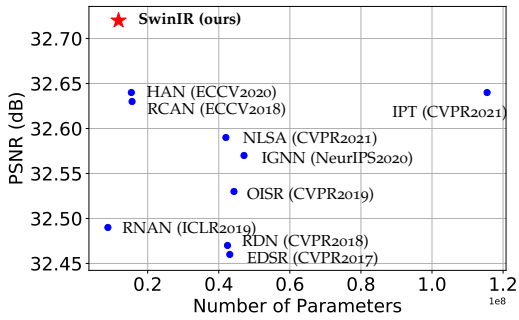



Figure 3.1: PSNR results v.s. the total number of parameters of different methods for image SR ($\times 4$) on Set5.

the local attention mechanism. On the other hand, it has the advantage of Transformer to model long-range dependency with the shifted window scheme. In this chapter, we propose an image restoration model, namely SwinIR, based on Swin Transformer. More specifically, SwinIR consists of three modules: shallow feature extraction, deep feature extraction and high-quality image reconstruction modules. Shallow feature extraction module uses a convolution layer to extract shallow feature, which is directly transmitted to the reconstruction module so as to preserve low-frequency information. Deep feature extraction module is mainly composed of residual Swin Transformer blocks (RSTB), each of which utilizes several Swin Transformer layers for local attention and cross-window interaction. In addition, we add a convolution layer at the end of the block for feature enhancement and use a residual connection to provide a shortcut for feature aggregation. Finally, both shallow and deep features are fused in the reconstruction module for high-quality image reconstruction.

Compared with prevalent CNN-based image restoration models, Transformer-based SwinIR has several benefits: (1) content-based interactions between image content and attention weights, which can be interpreted as spatially varying convolution [106]–[108]. (2) long-range dependency modelling are enabled by the shifted window mechanism. (3) better performance with less parameters. For example, as shown in Fig. 3.1, SwinIR achieves better PSNR with less parameters compared with existing image SR methods.

## 3.2 RELATED WORK

### 3.2.1 *Image Restoration*

Existing image restoration methods can be roughly divided into model-based [35], [38], [55], [96], [109] and learning-based methods [40], [48], [61], [93], [110], [111]. Due to page limit, we focus on learning-based methods on three tasks: image SR, image denoising and compression artifact reduction (JPEG image deblocking). Most of them learn mappings between low-quality and high-quality images from large-scale paired datasets.

IMAGE SUPER-RESOLUTION. Imgae super-resolution (SR) aims to reconstruct the high-resolution image from its low-resolution counterpart [38], [96], [109]. Dong *et al*. [17] proposed SRCNN that utilizes CNNs for image SR. Kim *et al*. [93] proposed a very deep CNN network VDSR based on the VGG [112]. Similarly, based the ResNet [97], Ledig *et al*. [49] used residual design in SRResNet, which was further enhanced by EDSR [39]. Later on, based on above pioneering work, more effective architectures were proposed to improve performance by using larger and deeper architectures, such as Laplacian pyramid [41], densely connected block [113], residual dense block [48], [111], back projection network [114], residual-in-residual dense block [40], residual channel attention block [18], second-order attention [115], graph aggregation module [116], holistic attention [117] and non-local sparse attention [118].

Specially, some works have exploited the attention mechanism [99] inside the convolution neural network framework. Zhang *et al*. [18] proposed a residual channel attention network that focuses on more informative channels by channel attention. Based on [18], Dai *et al*. [115] proposed a second-order attention network that refines features adaptively with second-order statistics, whereas Niu *et al*. [117] proposed a holistic attention network that models the interdependencies among positions, channels and network layers. Liu *et al*. [119] proposed non-local attention to explore long-range feature correlations, which was further improved by [120], [121] and [118]. The non-local attention mechanism is closely related to the proposed SwinIR, in which the self-attention mechanism can be interpreted as a specific instantiation of non-local means [106]. In addition, Zhou *et al*. [116] proposed an internal graph

network to aggregate HR patch features, which can be seen as a special case of spatial attention.

IMAGE DENOISING. Image denoising aims to restore images corrupted by noises. Zhang *et al*. [19] proposed a deep neural network DnCNN for image denoising. Since this milestone work, a flurry of CNN-based models have been proposed. Mao *et al*. [122] proposed an auto-encoder network with skip connections. Tai *et al*. [113] proposed to mine persistent memory by a recursive unit. Zhang *et al*. [20] proposed a fast and flexible model for multi-level denoising. There are other attempts on non-local modules [119], [120], [123], dilated residual block [124], residual dense block [111], optical control [125], *etc*.

COMPRESSION ARTIFACT REDUCTION. Compression artifact reduction aims to remove the artifacts generated by lossy compression. For example, JPEG compression will result in blocking, ringing and blurring artifacts [23]. In this case, the task is often referred to as JPEG image deblocking. Inspired by SRCNN [17], Dong *et al*. [23] proposed the a learnable artifact removal model ARCNN. Zhang *et al*. [19] proposed a very deep architecture DnCNN with residual learning. Later, more effective architectures are proposed to solve the problem, such as memory block [113], hierarchical skip connection [110], residual dense block [111], [120], multi-level wavelet-CNN [126] and residual U-Net [61]. Besides, there are some methods that utilize the low-level JPEG primitives for better performance [127]–[130].

### 3.2.2 *Vision Transformer*

Recently, natural language processing model Transformer [99] has gained much popularity in the computer vision community. When used in vision problems such as image classification [102], [103], [105], [106], [131]–[133], object detection [100], [101], [103], [134], segmentation [103], [131], [135], [136] and crowd counting [137], [138], it learns to attend to important image regions by exploring the global interactions between different regions. Due to its impressive performance, Transformer has also been introduced for image restoration [8], [104], [139]. Chen *et al*. [104] proposed a backbone model IPT for various restoration problems based on the standard Transformer. However, IPT relies on large number of parameters (over 115.5M parameters),

large-scale datasets (over 1.1M images) and multi-task learning for good performance. Cao *et al*. [8] proposed VSR-Transformer that uses the self-attention mechanism for better feature fusion in video SR, but image features are still extracted from CNN. Besides, both IPT and VSR-Transformer are patch-wise attention, which may be improper for image restoration. In addition, a concurrent work [139] proposed a U-shaped architecture based on the Swin Transformer [103]. However, the down-sampling operations make the multi-scale U-shaped design unreliable in preserving image details, which are crucial for some tasks such as image SR.

## 3.3 METHODOLOGY

In this section, we introduce the proposed SwinIR model. We first show the overall model architecture and then describe the key residual Swin Transformer block in detail.

### 3.3.1 *Model Architecture*

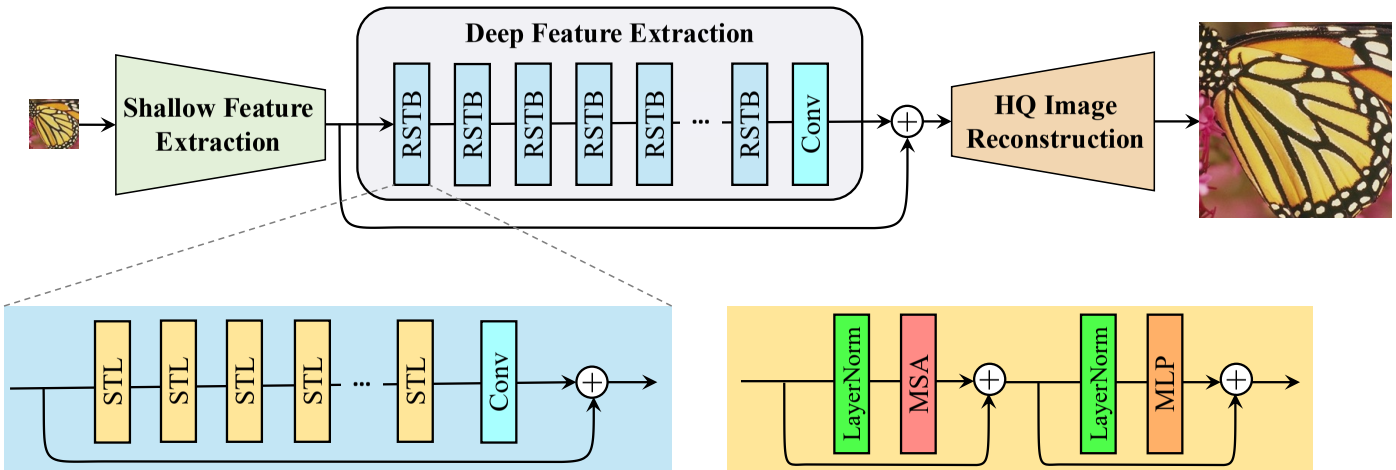
As shown in Fig. 3.2, SwinIR consists of three modules: shallow feature extraction, deep feature extraction and high-quality (HQ) image reconstruction modules. We employ the same feature extraction modules for all restoration tasks, but use different reconstruction modules for different tasks.

SHALLOW AND DEEP FEATURE EXTRACTION. Given a low-quality (LQ) input $I_{LQ} \in \mathbb{R}^{H\times W\times C_{in}}$ ($H$, $W$ and $C_{in}$ are the image height, width and input channel number, respectively), we use a $3 \times 3$ convolutional layer $H_{SF}(\cdot)$ to extract shallow feature $F_0 \in \mathbb{R}^{H\times W\times C}$ as

$$F_0 = H_{SF}(I_{LQ}), \tag{3.1}$$

where $C$ is the feature channel number. The convolution layer is good at early visual processing, leading to more stable optimization and better results [140]. It also provides a simple way to map the input image space to a higher dimensional feature space. Then, we extract deep feature $F_{DF} \in \mathbb{R}^{H\times W\times C}$ from $F_0$ as

$$F_{DF} = H_{DF}(F_0), \tag{3.2}$$




(a) RSTB (b) STL

Figure 3.2: The architecture of the proposed SwinIR for image restoration. It consists of three parts: shallow feature extraction, deep feature extraction and high-quality image reconstruction. The shallow feature extraction module extracts the shallow feature from the input low-quality image, while the deep feature extraction module further refine the features with the proposed residual swim transformer blocks (RSTB, see subfigure (a)). In the end, both shallow and deep features are added together to reconstruct the high-quality image.

where $H_{DF}(\cdot)$ is the deep feature extraction module and it contains $K$ residual Swin Transformer blocks (RSTB) and a $3 \times 3$ convolutional layer. More specifically, intermediate features $F_1, F_2, \ldots, F_K$ and the output deep feature $F_{DF}$ are extracted block by block as

$$\begin{aligned} F_i &= H_{RSTB_i}(F_{i-1}), \quad i = 1, 2, \ldots, K, \\ F_{DF} &= H_{CONV}(F_K), \end{aligned} \tag{3.3}$$

where $H_{RSTB_i}(\cdot)$ denotes the $i$-th RSTB and $H_{CONV}$ is the last convolutional layer. Using a convolutional layer at the end of feature extraction can bring the inductive bias of the convolution operation into the Transformer-based network, and lay a better foundation for the later aggregation of shallow and deep features.

IMAGE RECONSTRUCTION. Taking image SR as an example, we reconstruct the high-quality image $I_{RHQ}$ by aggregating shallow and deep features as

$$I_{RHQ} = H_{REC}(F_0 + F_{DF}), \tag{3.4}$$

where $H_{REC}(\cdot)$ is the function of the reconstruction module. Shallow feature mainly contain low-frequencies, while deep feature focus on recovering lost high-frequencies. With a long skip connection, SwinIR can transmit the low-frequency information directly to the reconstruction module, which can help deep feature extraction module focus on high-frequency information and stabilize training. For the implementation of reconstruction module, we use the sub-pixel convolution layer [141] to upsample the feature.

For tasks that do not need upsampling, such as image denoising and JPEG compression artifact reduction, a single convolution layer is used for reconstruction. Besides, we use residual learning to reconstruct the residual between the LQ and the HQ image instead of the HQ image. This is formulated as

$$I_{RHQ} = H_{SwinIR}(I_{LQ}) + I_{LQ}, \tag{3.5}$$

where $H_{SwinIR}(\cdot)$ denotes the function of SwinIR.

LOSS FUNCTION. For image SR, we optimize the parameters of SwinIR by minimizing the $L_1$ pixel loss

$$\mathcal{L} = \| I_{RHQ} - I_{HQ} \|_1, \tag{3.6}$$

where $I_{RHQ}$ is obtained by taking $I_{LQ}$ as the input of SwinIR, and $I_{HQ}$ is the corresponding ground-truth HQ image. For classical and lightweight image SR, we only use the naive $L_1$ pixel loss as same as previous work to show the effectiveness of the proposed network. For real-world image SR, we use a combination of pixel loss, GAN loss and perceptual loss [4], [40], [142]–[144] to improve visual quality.

For image denoising and JPEG compression artifact reduction, we use the Charbonnier loss [145]

$$\mathcal{L} = \sqrt{\|I_{RHQ} - I_{HQ}\|^2 + \epsilon^2}, \tag{3.7}$$

where $\epsilon$ is a constant that is empirically set to $10^{-3}$.

### 3.3.2 *Residual Swin Transformer Block*

As shown in Fig. 3.2(a), the residual Swin Transformer block (RSTB) is a residual block with Swin Transformer layers (STL) and convolutional layers. Given the input feature $F_{i,0}$ of the $i$-th RSTB, we first extract intermediate features $F_{i,1}, F_{i,2}, \ldots, F_{i,L}$ by $L$ Swin Transformer layers as

$$F_{i,j} = H_{STL_{i,j}}(F_{i,j-1}), \quad j = 1, 2, \ldots, L, \tag{3.8}$$

where $H_{STL_{i,j}}(\cdot)$ is the $j$-th Swin Transformer layer in the $i$-th RSTB. Then, we add a convolutional layer before the residual connection. The output of RSTB is formulated as

$$F_{i,out} = H_{CONV_i}(F_{i,L}) + F_{i,0}, \tag{3.9}$$

where $H_{CONV_i}(\cdot)$ is the convolutional layer in the $i$-th RSTB. This design has two benefits. First, although Transformer can be viewed as a specific instantiation of spatially varying convolution [106], [108], covolutional layers with spatially invariant filters can enhance the translational equivariance of SwinIR. Second, the residual connection provides a identity-based connection from different blocks to the reconstruction module, allowing the aggregation of different levels of features.

SWIN TRANSFORMER LAYER. Swin Transformer layer (STL) [103] is based on the standard multi-head self-attention of the original Transformer layer [99]. The main differences lie in local attention and the shifted window mechanism. As shown in Fig. 3.2(b), given an input

of size $H \times W \times C$, Swin Transformer first reshapes the input to a $\frac{HW}{M^2} \times M^2 \times C$ feature by partitioning the input into non-overlapping $M \times M$ local windows, where $\frac{HW}{M^2}$ is the total number of windows. Then, it computes the standard self-attention separately for each window (*i.e.*, local attention). For a local window feature $X \in \mathbb{R}^{M^2 \times C}$, the *query*, *key* and *value* matrices $Q$, $K$ and $V$ are computed as

$$Q = XP_Q, \quad K = XP_K, \quad V = XP_V, \tag{3.10}$$

where $P_Q$, $P_K$ and $P_V$ are projection matrices that are shared across different windows. Generally, we have $Q, K, V \in \mathbb{R}^{M^2 \times d}$. The attention matrix is thus computed by the self-attention mechanism in a local window as

$$\text{Attention}(Q, K, V) = \text{SoftMax}(QK^T/\sqrt{d} + B)V, \tag{3.11}$$

where $B$ is the learnable relative positional encoding. In practice, following [99], we perform the attention function for $h$ times in parallel and concatenate the results for multi-head self-attention (MSA).

Next, a multi-layer perceptron (MLP) that has two fully-connected layers with GELU non-linearity between them is used for further feature transformations. The LayerNorm (LN) layer is added before both MSA and MLP, and the residual connection is employed for both modules. The whole process is formulated as

$$\begin{aligned} X &= \text{MSA}(\text{LN}(X)) + X, \\ X &= \text{MLP}(\text{LN}(X)) + X. \end{aligned} \tag{3.12}$$

However, when the partition is fixed for different layers, there is no connection across local windows. Therefore, regular and shifted window partitioning are used alternately to enable cross-window connections [103], where shifted window partitioning means shifting the feature by $(\lfloor \frac{M}{2} \rfloor, \lfloor \frac{M}{2} \rfloor)$ pixels before partitioning.

## 3.4 EXPERIMENTS

### 3.4.1 *Experimental Setup*

We conduct experiments on three representative image restoration tasks, including image SR, image denoising and compression artifact

reduction, to evaluate the performance of the proposed model. In particular, for image SR, we first evaluate it on classical image SR and lightweight image SR, and then train the model with the proposed practical degradation model for real-world image SR.

ARCHITECTURE. For classical image SR, real-world image SR, image denoising and JPEG compression artifact reduction, the RSTB number, STL number, window size, channel number and attention head number are generally set to 6, 6, 8, 180 and 6, respectively. One exception is that the window size is set to 7 for JPEG compression artifact reduction, as we observe significant performance drop when using 8, possibly because JPEG encoding uses $8 \times 8$ image partions. For lightweight image SR, we decrease RSTB number and channel number to 4 and 60, respectively. Following [18], [117], when self-ensemble strategy [39] is used in testing, we mark the model with a symbol "+", *e.g.*, SwinIR+.

TRAINING. For classical and lightweight image SR, following [18], [117], [118], we train SwinIR on 800 training images of DIV2K [81]. Some compared methods (*e.g.*, [114], [40]) further use 2560 images from Flickr2K [146] for training, so we also train SwinIR on larger datasets (DIV2K+Flickr2K) to investigate whether SwinIR can further improve its performance. For fair comparison, we use $48 \times 48$ and $64 \times 64$ LQ image patches respectively in above two cases following the common settings. The HQ-LQ image pairs are obtained by the MATLAB bicubic kernel. The total training iterations and mini-batch size are set to 500K and 32, respectively. The learning rate is initialized as 2e-4 and reduced by half at [250K,400K,450K,475K]. For $\times 3$, $\times 4$ and $\times 8$ classical image SR, we initialize the model with $\times 2$ weights and halve the learning rate as well as total training iterations. Unlike other Transformer-based models that often uses AdamW [147] optimizer with cosine learning rate decay strategy, we find that using Adam [148] optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$ leads to better performance.

For real-world image SR, we use the same image degradation model as BSRGAN [4] and train it on a combination of DIV2K, Flickr2K and OST [149]. The model is trained for 1,000K iterations for the PSNR training stage. The learning rate is halved at [500K,800K,900K,950K]. For the GAN training stage, we train it for 600K iterations and the learning rate is halved at [400K,500K,550K,575K]. Weighting parameters between $L_1$ pixel loss, perceptual loss and GAN loss are 1, 1 and 0.1, respectively.

Note that we use the same EMA strategy, USM strategy, perceptual loss and GAN loss as [142].

For denoising and compression artifact reduction, following [61], [111], we use random crops from the combination of 800 DIV2K images, 2650 Flickr2K images, 400 BSD500 images [150] and 4744 WED images [82]. The batch size is 8. The patch sizes are $128 \times 128$ (window size is $8 \times 8$) and $126 \times 126$ (window size is $7 \times 7$), respectively. We obtain noisy images by adding additive white Gaussian noises (AWGN) with noise level $\sigma$, and compressed images by the MATLAB JPEG encoder with JPEG level $q$. The total training iterations and mini-batch size are set to 1600K and 8, respectively. The learning rate is halved at [800K,1200K,1400K,1500K]. When $\sigma = 15$ or $q = 40$, we train the model from scratch. When $\sigma = 25/50$ or $q = 10/20/30$, we fine-tune from $\sigma = 15$ or $q = 40$. Other details are the same as classical SR.

EVALUATION. Following the tradition of image SR, we report PSNR and SSIM [151] on the Y channel of the YCbCr space. For image denoising, we report the PSNR on the RGB channel and Y channel for color and grayscale denoising, respectively. For compression artifact reduction, in addtion to the Y channel PSNR and SSIM, we also report PNSR-B [152] that is specially designed for deblocking quality assessment. Particularly, we pad the image in testing so that the image size is a multiple of window size. We also find that using a sliding window strategy [104] to crop the image into patches can further improve the PSNR by $0.02 \sim 0.03$dB at the cost of longer testing time, so we do not use it for comparison.

### 3.4.2 *Results on Synthetic Image Super-Resolution*

We compare SwinIR with classical image SR methods, which mianly focus on performance. We also compare it with lightweight image SR methods whose model sizes and computation complexity are restricted.

CLASSICAL IMAGE SUPER-RESOLUTION. Table 3.1 shows the quantitative comparisons between SwinIR (middle size) and state-of-the-art methods: DBPN [114], RCAN [18], RRDB [40], IGNN [116], HAN [117], NLSA [118] and IPT [104]. As one can see, when trained on DIV2K, SwinIR achieves best performance on almost all five benchmark datasets for all scale factors. The maximum PSNR gain reaches 0.26dB on

Table 3.1: Quantitative comparison (average PSNR/SSIM) with state-of-the-art methods for classical image SR.

| Method | Scale | Training Dataset | Set5 [153] | Set14 [154] | BSD100 [155] | Urban100 [156] | Manga109 [157] |
|---|---|---|---|---|---|---|---|
| RCAN [18] | ×2 | DIV2K | 38.27/0.9614 | 34.12/0.9216 | 32.41/0.9027 | 33.34/0.9384 | 39.44/0.9786 |
| IGNN [116] | ×2 | DIV2K | 38.24/0.9613 | 34.07/0.9217 | 32.41/0.9025 | 33.23/0.9383 | 39.35/0.9786 |
| HAN [117] | ×2 | DIV2K | 38.27/0.9614 | 34.16/0.9217 | 32.41/0.9027 | 33.35/0.9385 | 39.46/0.9785 |
| NLSA [118] | ×2 | DIV2K | 38.34/0.9618 | 34.08/0.9231 | 32.43/0.9027 | 33.42/0.9394 | 39.59/0.9789 |
| **SwinIR (ours)** | ×2 | DIV2K | 38.35/0.9620 | 34.14/0.9227 | 32.44/0.9030 | 33.40/0.9393 | 39.60/0.9792 |
| **SwinIR+ (ours)** | ×2 | DIV2K | 38.38/0.9621 | 34.24/0.9233 | 32.47/0.9032 | 33.51/0.9401 | 39.70/0.9794 |
| DBPN [114] | ×2 | DIV2K+Flickr2K | 38.09/0.9600 | 33.85/0.9190 | 32.27/0.9000 | 32.55/0.9324 | 38.89/0.9775 |
| IPT [104] | ×2 | ImageNet | 38.37/- | 34.43/- | 32.48/- | 33.76/- | -/- |
| **SwinIR (ours)** | ×2 | DIV2K+Flickr2K | 38.42/0.9623 | 34.46/0.9250 | 32.53/0.9041 | 33.81/0.9427 | 39.92/0.9797 |
| **SwinIR+ (ours)** | ×2 | DIV2K+Flickr2K | 38.46/0.9624 | 34.61/0.9260 | 32.55/0.9043 | 33.95/0.9433 | 40.02/0.9800 |
| RCAN [18] | ×3 | DIV2K | 34.74/0.9299 | 30.65/0.8482 | 29.32/0.8111 | 29.09/0.8702 | 34.44/0.9499 |
| IGNN [116] | ×3 | DIV2K | 34.72/0.9298 | 30.66/0.8484 | 29.31/0.8105 | 29.03/0.8696 | 34.39/0.9496 |
| HAN [117] | ×3 | DIV2K | 34.75/0.9299 | 30.67/0.8483 | 29.32/0.8110 | 29.10/0.8705 | 34.48/0.9500 |
| NLSA [118] | ×3 | DIV2K | 34.85/0.9306 | 30.70/0.8485 | 29.34/0.8117 | 29.25/0.8726 | 34.57/0.9508 |
| **SwinIR (ours)** | ×3 | DIV2K | 34.89/0.9312 | 30.77/0.8503 | 29.37/0.8124 | 29.29/0.8744 | 34.74/0.9518 |
| **SwinIR+ (ours)** | ×3 | DIV2K | 34.95/0.9316 | 30.83/0.8511 | 29.41/0.8130 | 29.42/0.8761 | 34.92/0.9526 |
| IPT [104] | ×3 | ImageNet | 34.81/- | 30.85/- | 29.38/- | 29.49/- | -/- |
| **SwinIR (ours)** | ×3 | DIV2K+Flickr2K | 34.97/0.9318 | 30.93/0.8534 | 29.46/0.8145 | 29.75/0.8826 | 35.12/0.9537 |
| **SwinIR+ (ours)** | ×3 | DIV2K+Flickr2K | 35.04/0.9322 | 31.00/0.8542 | 29.49/0.8150 | 29.90/0.8841 | 35.28/0.9543 |
| RCAN [18] | ×4 | DIV2K | 32.63/0.9002 | 28.87/0.7889 | 27.77/0.7436 | 26.82/0.8087 | 31.22/0.9173 |
| IGNN [116] | ×4 | DIV2K | 32.57/0.8998 | 28.85/0.7891 | 27.77/0.7434 | 26.84/0.8090 | 31.28/0.9182 |
| HAN [117] | ×4 | DIV2K | 32.64/0.9002 | 28.90/0.7890 | 27.80/0.7442 | 26.85/0.8094 | 31.42/0.9177 |
| NLSA [118] | ×4 | DIV2K | 32.59/0.9000 | 28.87/0.7891 | 27.78/0.7444 | 26.96/0.8109 | 31.27/0.9184 |
| **SwinIR (ours)** | ×4 | DIV2K | 32.72/0.9021 | 28.94/0.7914 | 27.83/0.7459 | 27.07/0.8164 | 31.67/0.9226 |
| **SwinIR+ (ours)** | ×4 | DIV2K | 32.81/0.9029 | 29.02/0.7928 | 27.87/0.7466 | 27.21/0.8187 | 31.88/0.9423 |
| DBPN [114] | ×4 | DIV2K+Flickr2K | 32.47/0.8980 | 28.82/0.7860 | 27.72/0.7400 | 26.38/0.7946 | 30.91/0.9137 |
| IPT [104] | ×4 | ImageNet | 32.64/- | 29.01/- | 27.82/- | 27.26/- | -/- |
| RRDB [40] | ×4 | DIV2K+Flickr2K | 32.73/0.9011 | 28.99/0.7917 | 27.85/0.7455 | 27.03/0.8153 | 31.66/0.9196 |
| **SwinIR (ours)** | ×4 | DIV2K+Flickr2K | 32.92/0.9044 | 29.09/0.7950 | 27.92/0.7489 | 27.45/0.8254 | 32.03/0.9260 |
| **SwinIR+ (ours)** | ×4 | DIV2K+Flickr2K | 32.93/0.9043 | 29.15/0.7958 | 27.95/0.7494 | 27.56/0.8273 | 32.22/0.9273 |

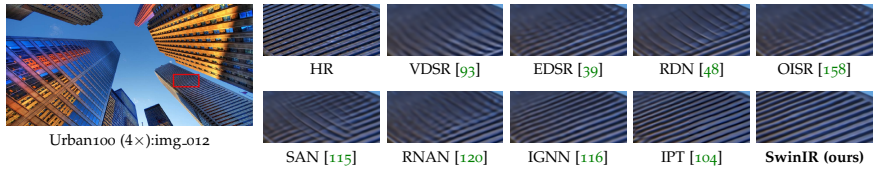



Figure 3.3: Visual comparison of bicubic image SR ($\times 4$) methods. Compared images are derived from [104]. Best viewed by zooming.

Manga109 for scale factor 4. Note that RCAN and HAN introduce channel and spatial attention, IGNN proposes adaptive patch feature aggregation, and NLSA is based on the non-local attention mechanism. However, all these CNN-based attention mechanisms perform worse than the proposed Transformer-based SwinIR, which indicates the effectiveness of the proposed model. When we train SwinIR on a larger dataset (DIV2K+Flickr2K), the performance further increases by a large margin (up to 0.47dB), achieving better accuracy than the same Transformer-based model IPT, even though IPT utilizes ImageNet (more than 1.3M images) in training and has huge number of parameters (115.5M). In contrast, SwinIR has a small number of parameters (11.8M) even compared with state-of-the-art CNN-based models (15.4$\sim$44.3M). As for runtime, representative CNN-based model RCAN, IPT and SwinIR take about 0.2, 4.5s and 1.1s to test on a $1,024 \times 1,024$ image, respectively.

Visual comparisons are show in Fig. 3.3. SwinIR can restore high-frequency details and alleviate the blurring artifacts, resulting in sharp and natural edges. In contrast, most CNN-based methods produces blurry images or even incorrect textures. IPT generates better images compared with CNN-based methods, but it suffers from image distortions and border artifact.

To compare the efficiency between convolutional models and SwinIR, we choose a representative model RCAN and train/test both models on 8 GeForce RTX2080Ti GPUs. As shown in Table 3.2, SwinIR beats RCAN with less number of parameters and FLOPs. Although it needs more runtime and larger testing memory, it brings a PSNR improvement of 0.45dB. For the discrepancy between runtime and FLOPs, we argue that it is because the attention operation in SwinIR consists of multiple sub-operations, which is less optimized compared with the convolution operation.

Table 3.2: Comparison of model size, training time, runtime, testing memory and FLOPs.

| Method | #Params | Training time | Runtime | Testing memory | #FLOPs | PSNR/SSIM |
|---|---|---|---|---|---|---|
| RCAN [18] | 15.6M | 1.6 days | 0.180s | 593.1M | 850.6G | 31.22/0.9173 |
| **SwinIR (ours)** | 11.9M | 1.8 days | 0.539s | 986.8M | 788.6G | 31.67/0.9226 |

LIGHTWEIGHT IMAGE SUPER-RESOLUTION. We also provide comparison of SwinIR (small size) with state-of-the-art lightweight image SR methods: CARN [159], FALSR-A [160], IMDN [42], LAPAR-A [161] and LatticeNet [162]. In addition to PSNR and SSIM, we also report the total numbers of parameters and multiply-accumulate operations (evaluated on a $1280 \times 720$ HQ image) to compare the model size and computational complexity of different models. As shown in Table 3.3, SwinIR outperforms competitive methods by a PSNR margin of up to 0.53dB on different benchmark datasets, with similar total numbers of parameters and multiply-accumulate operations. This indicates that the SwinIR architecture is highly efficient for image restoration.

### 3.4.3 *Results on Real-World Image Super-Resolution*

The ultimate goal of image SR is for real-world applications. In the last chapter, we proposed a practical degradation model BSRGAN for real-world image SR and achieved surprising results in real scenarios. To test the performance of SwinIR for real-world SR, we re-train SwinIR by using the same degradation model as BSRGAN for low-quality image synthesis. Since there is no ground-truth high-quality images, we only provide visual comparison with representative bicubic model ESRGAN [40] and state-of-the-art real-world image SR models RealSR [53], BSRGAN [4] and Real-ESRGAN [142]. As shown in Fig. 3.4, SwinIR produces visually pleasing images with clear and sharp edges, whereas other compared methods may suffer from unsatisfactory artifacts. In addition, to exploit the full potential of SwinIR for real applications, we further propose a large model and train it on much larger datasets. Experiments show that it can deal with more complex corruptions and achieves even better performance on real-world images than the current model.

Table 3.3: Quantitative comparison (average PSNR/SSIM) with state-of-the-art methods for lightweight image SR on benchmark datasets.

| Method | Scale | #Params | #Mult-Adds | Set5 [153] | Set14 [154] | BSD100 [155] | Urban100 [156] | Manga109 [157] |
|---|---|---|---|---|---|---|---|---|
| CARN [159] | ×2 | 1,592K | 222.8G | 37.76/0.9590 | 33.52/0.9166 | 32.09/0.8978 | 31.92/0.9256 | 38.36/0.9765 |
| FALSR-A [160] | ×2 | 1,021K | 234.7G | 37.82/0.959 | 33.55/0.9168 | 32.10/0.8987 | 31.93/0.9256 | -/- |
| IMDN [42] | ×2 | 694K | 158.8G | 38.00/0.9605 | 33.63/0.9177 | 32.19/0.8996 | 32.17/0.9283 | 38.88/0.9774 |
| LAPAR-A [161] | ×2 | 548K | 171.0G | 38.01/0.9605 | 33.62/0.9183 | 32.19/0.8999 | 32.10/0.9283 | 38.67/0.9772 |
| LatticeNet [162] | ×2 | 756K | 169.5G | 38.15/0.9610 | 33.78/0.9193 | 32.25/0.9005 | 32.43/0.9302 | -/- |
| **SwinIR (ours)** | ×2 | 878K | 195.6G | 38.14/0.9611 | 33.86/0.9206 | 32.31/0.9012 | 32.76/0.9340 | 39.12/0.9783 |
| CARN [159] | ×3 | 1,592K | 118.8G | 34.29/0.9255 | 30.29/0.8407 | 29.06/0.8034 | 28.06/0.8493 | 33.50/0.9440 |
| IMDN [42] | ×3 | 703K | 71.5G | 34.36/0.9270 | 30.32/0.8417 | 29.09/0.8046 | 28.17/0.8519 | 33.61/0.9445 |
| LAPAR-A [161] | ×3 | 544K | 114.0G | 34.36/0.9267 | 30.34/0.8421 | 29.11/0.8054 | 28.15/0.8523 | 33.51/0.9441 |
| LatticeNet [162] | ×3 | 765K | 76.3G | 34.53/0.9281 | 30.39/0.8424 | 29.15/0.8059 | 28.33/0.8538 | -/- |
| **SwinIR (ours)** | ×3 | 886K | 87.2G | 34.62/0.9289 | 30.54/0.8463 | 29.20/0.8082 | 28.66/0.8624 | 33.98/0.9478 |
| CARN [159] | ×4 | 1,592K | 90.9G | 32.13/0.8937 | 28.60/0.7806 | 27.58/0.7349 | 26.07/0.7837 | 30.47/0.9084 |
| IMDN [42] | ×4 | 715K | 40.9G | 32.21/0.8948 | 28.58/0.7811 | 27.56/0.7353 | 26.04/0.7838 | 30.45/0.9075 |
| LAPAR-A [161] | ×4 | 659K | 94.0G | 32.15/0.8944 | 28.61/0.7818 | 27.61/0.7366 | 26.14/0.7871 | 30.42/0.9074 |
| LatticeNet [162] | ×4 | 777K | 43.6G | 32.30/0.8962 | 28.68/0.7830 | 27.62/0.7367 | 26.25/0.7873 | -/- |
| **SwinIR (ours)** | ×4 | 897K | 49.6G | 32.44/0.8976 | 28.77/0.7858 | 27.69/0.7406 | 26.47/0.7980 | 30.92/0.9151 |

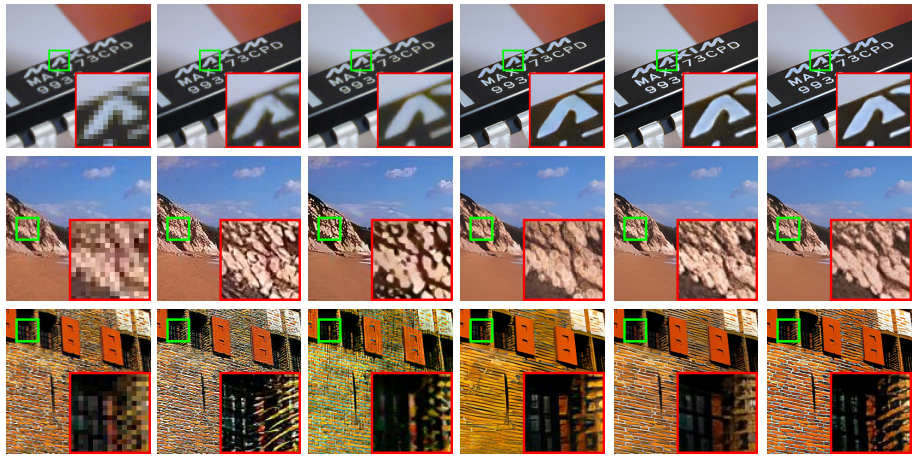


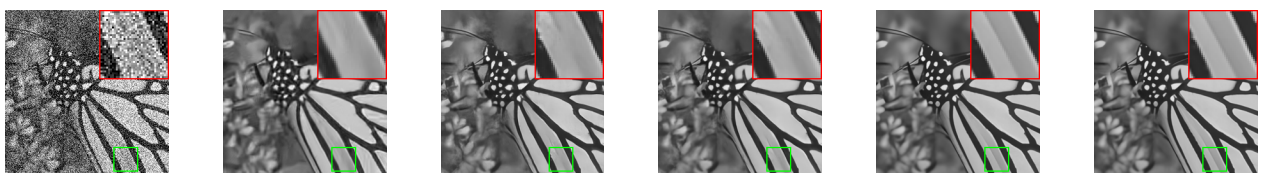

Figure 3.4: Visual comparison of real-world image SR (×4) methods on real-world images.

### 3.4.4 *Results on Image Denoising*

We show grayscale and color image denoising results in Table 3.4 and Table 3.5, respectively. Compared methods include traditional models BM3D [95] and WNNM [163], CNN-based models DnCNN [19], IRCNN [58], FFDNet [20], N3Net [123], NLRN [119], FOCNet [125], RNAN [120], MWCNN [126] and DRUNet [61]. Following [19], [61], the compared noise levels include 15, 25 and 50. As one can see, our model achieves better performance than all compared methods. In particular, it surpasses the state-of-the-art model DRUNet by up to 0.3dB on the large Urban100 dataset that has 100 high-resolution testing images. It is worth pointing out that SwinIR only has 12.0M parameters, whereas DRUNet has 32.7M parameters. This indicates that the SwinIR architecture is highly efficient in learning feature representations for restoration. The visual comparison for grayscale and color image denoising of different methods are shown in Figs. 3.5 and 3.6. As we can see, our method can remove heavy noise corruption and preserve high-frequency image details, resulting in sharper edges and more natural textures. By contrast, other methods suffer from either over-smoothness or over-sharpness, and cannot recover rich textures.

Table 3.4: Quantitative comparison (average PSNR) with state-of-the-art methods for grayscale image denoising on benchmark datasets.

| Dataset | $\sigma$ | BM3D [95] | WNNM [163] | DnCNN [19] | IRCNN [58] | FFDNet [20] | N3Net [123] | NLRN [119] | FOCNet [125] | RNAN [120] | MWCNN [126] | DRUNet [61] | **SwinIR (ours)** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Set12 [19] | 15 | 32.37 | 32.70 | 32.86 | 32.76 | 32.75 | - | 33.16 | 33.07 | - | 33.15 | 33.25 | 33.36 |
| | 25 | 29.97 | 30.28 | 30.44 | 30.37 | 30.43 | 30.55 | 30.80 | 30.73 | - | 30.79 | 30.94 | 31.01 |
| | 50 | 26.72 | 27.05 | 27.18 | 27.12 | 27.32 | 27.43 | 27.64 | 27.68 | 27.70 | 27.74 | 27.90 | 27.91 |
| BSD68 [164] | 15 | 31.08 | 31.37 | 31.73 | 31.63 | 31.63 | - | 31.88 | 31.83 | - | 31.86 | 31.91 | 31.97 |
| | 25 | 28.57 | 28.83 | 29.23 | 29.15 | 29.19 | 29.30 | 29.41 | 29.38 | - | 29.41 | 29.48 | 29.50 |
| | 50 | 25.60 | 25.87 | 26.23 | 26.19 | 26.29 | 26.39 | 26.47 | 26.50 | 26.48 | 26.53 | 26.59 | 26.58 |
| Urban100 [156] | 15 | 32.35 | 32.97 | 32.64 | 32.46 | 32.40 | - | 33.45 | 33.15 | - | 33.17 | 33.44 | 33.70 |
| | 25 | 29.70 | 30.39 | 29.95 | 29.80 | 29.90 | 30.19 | 30.94 | 30.64 | - | 30.66 | 31.11 | 31.30 |
| | 50 | 25.95 | 26.83 | 26.26 | 26.22 | 26.50 | 26.82 | 27.49 | 27.40 | 27.65 | 27.42 | 27.96 | 27.98 |



Noisy | BM3D [95] | DnCNN [19] | FFDNet [20] | DRUNet [61] | **SwinIR (ours)**

Figure 3.5: Visual comparison of grayscale image denoising (noise level 50) methods on image *"Monarch"* from Set12 [19]. Compared images are derived from [61].

Table 3.5: Quantitative comparison (average PSNR) with state-of-the-art methods for color image denoising on benchmark datasets.

| Dataset | $\sigma$ | BM3D [95] | DnCNN [19] | IRCNN [58] | FFDNet [20] | DSNet [124] | RPCNN [165] | BRDNet [166] | RNAN [120] | RDN [111] | IPT [104] | DRUNet [61] | **SwinIR (ours)** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CBSD68 [164] | 15 | 33.52 | 33.90 | 33.86 | 33.87 | 33.91 | - | 34.10 | - | - | - | 34.30 | 34.42 |
| | 25 | 30.71 | 31.24 | 31.16 | 31.21 | 31.28 | 31.24 | 31.43 | - | - | - | 31.69 | 31.78 |
| | 50 | 27.38 | 27.95 | 27.86 | 27.96 | 28.05 | 28.06 | 28.16 | 28.27 | 28.31 | 28.39 | 28.51 | 28.56 |
| Kodak24 [167] | 15 | 34.28 | 34.60 | 34.69 | 34.63 | 34.63 | - | 34.88 | - | - | - | 35.31 | 35.34 |
| | 25 | 32.15 | 32.14 | 32.18 | 32.13 | 32.16 | 32.34 | 32.41 | - | - | - | 32.89 | 32.89 |
| | 50 | 28.46 | 28.95 | 28.93 | 28.98 | 29.05 | 29.25 | 29.22 | 29.58 | 29.66 | 29.64 | 29.86 | 29.79 |
| McMaster [168] | 15 | 34.06 | 33.45 | 34.58 | 34.66 | 34.67 | - | 35.08 | - | - | - | 35.40 | 35.61 |
| | 25 | 31.66 | 31.52 | 32.18 | 32.35 | 32.40 | 32.33 | 32.75 | - | - | - | 33.14 | 33.20 |
| | 50 | 28.51 | 28.62 | 28.91 | 29.18 | 29.28 | 29.33 | 29.52 | 29.72 | - | 29.98 | 30.08 | 30.22 |
| Urban100 [156] | 15 | 33.93 | 32.98 | 33.78 | 33.83 | - | - | 34.42 | - | - | - | 34.81 | 35.13 |
| | 25 | 31.36 | 30.81 | 31.20 | 31.40 | - | 31.81 | 31.99 | - | - | - | 32.60 | 32.90 |
| | 50 | 27.93 | 27.59 | 27.70 | 28.05 | - | 28.62 | 28.56 | 29.08 | 29.38 | 29.71 | 29.61 | 29.82 |

Noisy | DnCNN [19] | FFDNet [20] | IPT [104] | DRUNet [61] | **SwinIR (ours)**

Figure 3.6: Visual comparison of color image denoising (noise level 50) methods on image *"163085"* from CBSD68 [164]. Compared images are derived from [61].

Table 3.6: Quantitative comparison (average PSNR/SSIM/PSNR-B) with state-of-the-art methods for JPEG compression artifact reduction on benchmark datasets.

| Dataset | $q$ | ARCNN [23] | DnCNN-3 [19] | RNAN [120] |
|---|---|---|---|---|
| Classic5 [169] | 10 | 29.03/0.7929/28.76 | 29.40/0.8026/29.13 | 29.96/0.8178/29.62 |
| | 20 | 31.15/0.8517/30.59 | 31.63/0.8610/31.19 | 32.11/0.8693/31.57 |
| | 30 | 32.51/0.8806/31.98 | 32.91/0.8861/32.38 | 33.38/0.8924/32.68 |
| | 40 | 33.32/0.8953/32.79 | 33.77/0.9003/33.20 | 34.27/0.9061/33.4 |
| Dataset | $q$ | RDN [111] | DRUNet [61] | **SwinIR (ours)** |
| Classic5 [169] | 10 | 30.00/0.8188/- | 30.16/0.8234/29.81 | 30.27/0.8249/29.95 |
| | 20 | 32.15/0.8699/- | 32.39/0.8734/31.80 | 32.52/0.8748/31.99 |
| | 30 | 33.43/0.8930/- | 33.59/0.8949/32.82 | 33.73/0.8961/33.03 |
| | 40 | 34.27/0.9061/- | 34.41/0.9075/33.51 | 34.52/0.9082/33.66 |
| Dataset | $q$ | ARCNN [23] | DnCNN-3 [19] | RNAN [120] |
| LIVE1 [170] | 10 | 28.96/0.8076/28.77 | 29.19/0.8123/28.90 | 29.63/0.8239/29.25 |
| | 20 | 31.29/0.8733/30.79 | 31.59/0.8802/31.07 | 32.03/0.8877/31.44 |
| | 30 | 32.67/0.9043/32.22 | 32.98/0.9090/32.34 | 33.45/0.9149/32.71 |
| | 40 | 33.63/0.9198/33.14 | 33.96/0.9247/33.28 | 34.47/0.9299/33.66 |
| Dataset | $q$ | RDN [111] | DRUNet [61] | **SwinIR (ours)** |
| LIVE1 [170] | 10 | 29.67/0.8247/- | 29.79/0.8278/29.48 | 29.86/0.8287/29.50 |
| | 20 | 32.07/0.8882/- | 32.17/0.8899/31.69 | 32.25/0.8909/31.70 |
| | 30 | 33.51/0.9153/- | 33.59/0.9166/32.99 | 33.69/0.9174/33.01 |
| | 40 | 34.51/0.9302/- | 34.58/0.9312/33.93 | 34.67/0.9317/33.88 |

### 3.4.5 *Results on Compression Artifact Reduction*

Table 3.6 shows the comparison of SwinIR with state-of-the-art compression artifact reduction methods: ARCNN [23], DnCNN-3 [19], QGAC [130], RDN [111] and DRUNet [61]. All of compared methods are CNN-based models. Following [61], [111], we test different methods on two benchmark datasets (Classic5 [169] and LIVE1 [170]) for JPEG quality factors 10, 20, 30 and 40. As we can see, the proposed SwinIR has average PSNR gains of at least 0.11dB and 0.07dB on two testing datasets for different quality factors. Besides, compared with the previous best model DRUNet, SwinIR only has 11.5M parameters, while DRUNet is a large model that has 32.7M parameters.

### 3.4.6 *Ablation Study and Discussion*

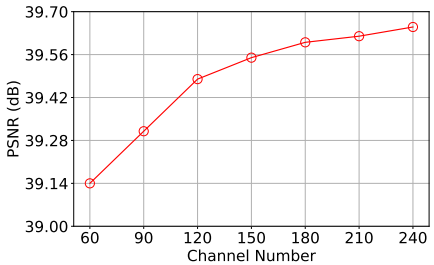
For ablation study, we train SwinIR on DIV2K [81] for classical image SR ($\times 2$) and test it on Manga109 [157].

IMPACT OF CHANNEL NUMBER, RSTB NUMBER AND STL NUMBER. We show the effects of channel number, RSTB number and STL number in a RSTB on model performance in Figs. 3.7a, 3.7b and 3.7c, respectively. It is observed that the PSNR is positively correlated with these three hyper-parameters. For channel number, although the performance keeps increasing, the total number of parameters grows quadratically. To balance the performance and model size, we choose 180 as the channel number in rest experiments. As for RSTB number and layer number, the performance gain becomes saturated gradually. We choose 6 for both of them to obtain a relatively small model.

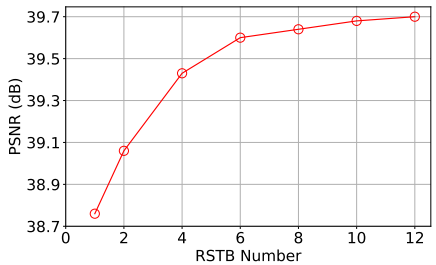
IMPACT OF PATCH SIZE AND TRAINING IMAGE NUMBER. We compare the proposed SwinIR with a representative CNN-based model RCAN to compare the difference between Transformer-based and CNN-based models. From Fig. 3.7d, one can see that SwinIR performs better than RCAN on different patch sizes, and the PSNR gain increases when the patch size increases. Fig. 3.7e shows the impact of the number of training images. Extra images from Flickr2K are used in training when the percentage is larger than 100% (800 images). There are two observations. First, as expected, the performance of SwinIR increases with the training image number. Second, different from the observation in IPT that Transformer-based models are heavily relied on large amount of training data, SwinIR achieves better results than CNN-based models using the same training data, even when the dataset is small (*i.e.*, 25%, 200 images).

MODEL CONVERGENCE COMPARISON. We also plot the PSNR during training for both SwinIR and RCAN in Fig. 3.7f. It is clear that SwinIR converges faster and better than RCAN, which is contradictory to previous observations that Transformer-based models often suffer from slow model convergence.

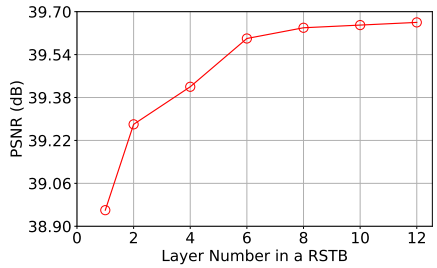
IMPACT OF RESIDUAL CONNECTION AND CONVOLUTION LAYER IN RSTB. Table 3.7 shows four residual connection variants in RSTB: no residual connection, using $1 \times 1$ convolution layer, using $3 \times 3$ convolution layer and using three $3 \times 3$ convolution layers (channel number of the intermediate layer is set to one fourth of network channel number). From the table, we can have following observations. First, the residual connection in RSTB is important as it improves the PSNR by 0.16dB. Sec-



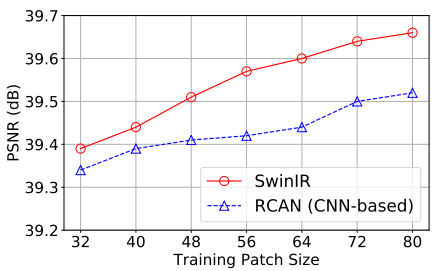

(a) PSNR *v.s.*Channel Number



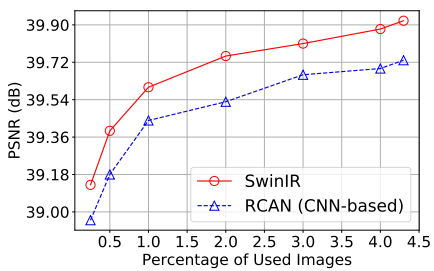

(b) PSNR *v.s.*RSTB Number



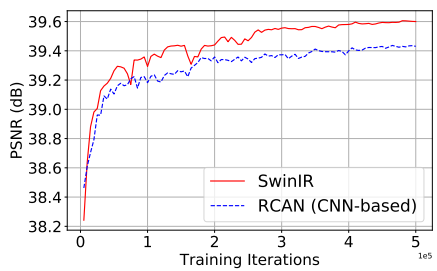

(c) PSNR *v.s.*Layer Number




(d) PSNR *v.s.*Patch Size




(e) PSNR *v.s.*Training Image Number




(f) PSNR *v.s.*Training Iteration

Figure 3.7: Ablation study on different settings of SwinIR. Results are tested on Manga109 [157] for image SR (×2).

Table 3.7: Ablation study on RSTB design.

| Design | No residual | $1 \times 1$ conv | $3 \times 3$ conv | Three $3 \times 3$ conv |
|---|---|---|---|---|
| PSNR | 39.42 | 39.45 | 39.58 | 39.56 |

ond, using $1 \times 1$ convolution brings little improvement maybe because it cannot extract local neighbouring information as $3 \times 3$ convolution does. Third, although using three $3 \times 3$ convolution layers can reduce the number of parameters, the performance drops slightly.

## 3.5 CONCLUSION

In this chapter, we proposed a transformer-based image restoration model SwinIR. The model is composed of three parts: shallow feature extraction, deep feature extraction and HR reconstruction modules. In particular, we use a stack of residual Swin Transformer blocks (RSTB) for deep feature extraction, and each RSTB is composed of Swin Transformer layers, convolution layer and a residual connection. Extensive experiments show that SwinIR achieves state-of-the-art performance on three representative image restoration tasks and six different settings: classic image SR, lightweight image SR, real-world image SR, grayscale image denoising, color image denoising and JPEG compression artifact reduction, which demonstrates the effectiveness and generalizability of the proposed SwinIR. In the future, we will extend the model to other restoration tasks such as image deblurring, deraining and dehazing.

# 4 VIDEO RESTORATION TRANSFORMER

In the last chapter, we tackled with one of the fundamental problems in low-level vision: single image restoration. In reality, with the increasing of bandwidth and storage medium, videos are becoming part of our daily life, leading to a widespread demand of video restoration algorithms. Different from single image restoration that restores a single high-quality image from a single low-quality input image, video restoration aims to restore multiple high-quality frames from multiple low-quality frames. Therefore, video restoration generally requires to utilize temporal information from multiple adjacent but usually misaligned video frames.

Existing deep methods generally tackle with this by exploiting a sliding window strategy or a recurrent architecture, which either is restricted by frame-by-frame restoration or lacks long-range modelling ability. In this chapter, we propose a Video Restoration Transformer (VRT) with parallel frame prediction and long-range temporal dependency modelling abilities. More specifically, VRT is composed of multiple scales, each of which consists of two kinds of modules: temporal reciprocal self attention (TRSA) and parallel warping. TRSA divides the video into small clips, on which reciprocal attention is applied for joint motion estimation, feature alignment and feature fusion, while self attention is used for feature extraction. To enable cross-clip interactions, the video sequence is shifted for every other layer. Besides, parallel warping is used to further fuse information from neighboring frames by parallel feature warping. Experimental results on five tasks, including video super-resolution, video deblurring, video denoising, video frame interpolation and space-time video super-resolution, demonstrate that VRT outperforms the state-of-the-art methods by large margins (up to 2.16dB) on fourteen benchmark datasets.

## 4.1 INTRODUCTION

Video restoration, which reconstructs high-quality (HQ) frames from multiple low-quality (LQ) frames, has attracted much attention recently.

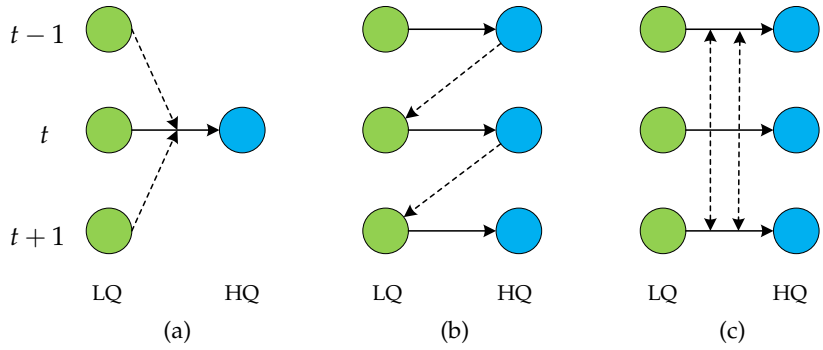



Figure 4.1: Illustrative comparison of sliding window-based models (4.1a, *e.g.*, [24], [171], [172]), recurrent models (4.1b, *e.g.*, [173]–[177]) and the proposed parallel VRT model (4.1c). Green and blue circles denote low-quality (LQ) input frames and high-quality (HQ) output frames, respectively. $t-1$, $t$ and $t+1$ are frame serial numbers. Dashed lines represent information fusion among different frames.

Compared with image restoration, the key challenge of video restoration lies in how to make full use of neighboring highly-related but misaligned supporting frames for reconstructing reference frames.

Existing video restoration methods can be mainly divided into two categories: sliding window-based methods [24], [171], [172], [178]–[183] and recurrent methods [173]–[177], [184]–[190]. As shown in Fig. 4.1a, sliding window-based methods generally input multiple frames to generate a single HQ frame and processes long video sequences in a sliding window fashion. Each input frame is processed for multiple times in inference, leading to inefficient feature utilization and increased computation cost.

Some other methods are based on a recurrent architecture. As shown in Fig. 4.1b, recurrent models mainly use previously reconstructed HQ frames for subsequent frame reconstruction. Due to the recurrent nature, they have three disadvantages. First, recurrent methods are limited in parallelization for efficient distributed training and inference. Second, although information is accumulated frame by frame, recurrent models are not good at long-range temporal dependency modelling. One frame may strongly affect the next adjacent frame, but its influence

is quickly lost after few time steps [99], [191]. Third, they suffer from significant performance drops on few-frame videos [8].

In this chapter, we propose a Video Restoration Transformer (VRT) that allows for parallel computation and long-range dependency modelling in video restoration. Based on a multi-scale framework, VRT divides the video sequence into non-overlapping clips and shifts it alternately to enable inter-clip interactions. Specifically, each scale of VRT has several temporal reciprocal self attention (TRSA) modules followed by a parallel warping module. In TRSA, reciprocal attention is focused on mutual alignment between neighboring two-frame clips, while self attention is used for feature extraction. At the end of each scale, we further use parallel warping to fuse neighboring frame information into the current frame. After multi-scale feature extraction, alignment and fusion, the HQ frames are individually reconstructed from their corresponding frame features.

Compared with existing video restoration frameworks, VRT has several benefits. First, as shown in Fig. 4.1c, VRT is trained and tested on long video sequences in parallel. In contrast, both sliding window-based and recurrent methods are often tested frame by frame. Second, VRT has the ability to model long-range temporal dependencies, utilizing information from multiple neighbouring frames during the reconstruction of each frame. By contrast, sliding window-based methods cannot be easily scaled up to long sequence modelling, while recurrent methods may forget distant information after several timestamps. Third, VRT proposes to use reciprocal attention for joint feature alignment and fusion. It adaptively utilizes features from supporting frames and fuses them into the reference frame, which can be regarded as implicit motion estimation and feature warping.

Our contributions can be summarized as follows:

1) We propose a new framework named Video Restoration Transformer (VRT) that is characterized by parallel computation and long-range dependency modelling. It jointly extracts, aligns, and fuses frame features at multiple scales.

2) We propose the reciprocal attention for mutual alignment between frames. It is a generalized "soft" version of image warping after implicit motion estimation.

3) VRT achieves state-of-the-art performance on video restoration, including video super-resolution, deblurring, denoising, frame in-

terpolation and space-time video super-resolution. It outperforms state-of-the-art methods by up to 2.16dB on benchmark datasets.

## 4.2 RELATED WORK

### 4.2.1 *Video Restoration*

Similar to image restoration [1], [4]–[7], [17]–[19], [48]–[50], [61], [69], [118]–[121], [192]–[200], learning-based methods, especially CNN-based methods, have become the primary workhorse for video restoration [24], [27], [176], [181], [201]–[211].

FRAMEWORK DESIGN. From the perspective of architecture design, existing methods can be roughly divided into two categories: sliding window-based and recurrent methods. Sliding window-based methods often takes a short sequence of frames as input and merely predict the center frame [24]–[26], [171], [172], [178]–[183], [212]. Although some works [213] predict multiple frames, they still focus on the reconstruction of the center frame during training and testing. Recurrent framework is another popular choice [173]–[177], [184]–[190]. Huang *et al*. [174] propose a bidirectional recurrent convolutional neural network for SR. Sajjadi *et al*. [187] warp the previous frame prediction onto the current frame and feed it to a restoration network along with the current input frame. This idea is used by Chan *et al*. [176] for bidirectional recurrent network, and further extended as grid propagation in [177].

TEMPORAL ALIGNMENT AND FUSION. Since supporting frames are often highly-related but misaligned, temporal alignment plays an critical role in video restoration [24], [172], [176], [177], [214]–[216]. Early methods [178], [214], [217]–[219] use traditional flow estimation methods to estimate optical flow and warp the supporting frames towards the reference frame. To compensate occlusion and large motion, Xue *et al*. [215] utilize task-oriented flow by fine-tuning the pre-trained optical flow estimation model SpyNet [220] on different video restoration tasks. Jo *et al*. [221] use dynamic upsampling filters for implicit motion compensation. Kim *et al*. [222] propose a spatio-temporal transformer network for multi-frame optical flow estimation and warping. Tian *et al*. [172] propose TDAN that utilize deformable convolution [223] for feature alignment. Based on TDAN, Wang *et al*. [24] extend it to multi-

scale alignment, while Chan *et al*. [177] incorporate optical flow as a guidance for offsets learning.

ATTENTION MECHANISM. Attention mechanism has been exploited in video restoration in combination with CNN [8], [24], [28], [218]. Liu *et al*. [218] learn different weights for different temporal branches. Wang *et al*. [24] learn pixel-level attention maps for spatial and temporal feature fusion. To better incorporate temporal information, Isobe *et al*. [182] divide frames into several groups and design a temporal group attention module. Suin *et al*. [28] propose a reinforcement learning-based framework with factorized spatio-temporal attention. Cao *et al*. [8] propose to use self attention among local patches within a video.

### 4.2.2 *Vision Transformer*

Recently, Transformer-based models [99], [224]–[226] have achieved promising performance in various vision tasks, such as image recognition [100], [102], [103], [132]–[134], [138], [226], [227] and image restoration [1], [104], [139]. Some methods have tried to use Transformer for video modelling by extending the attention mechanism to the temporal dimension [225], [228]–[231]. However, most of them are designed for visual recognition, which are fundamentally different from restoration tasks. They are more focused on feature fusion than on alignment. Cao *et al*. [8] propose a CNN-transformer hybrid network for video super-resolution (SR) based on spatial-temporal convolutional self attention. However, it does not make full use of local information within each patch and suffers from border artifacts during testing.

## 4.3 METHODOLOGY

### 4.3.1 *Overall Framework*

Let $I^{LQ} \in \mathbb{R}^{T\times H\times W\times C_{in}}$ be a sequence of low-quality (LQ) input frames and $I^{HQ} \in \mathbb{R}^{T\times sH\times sW\times C_{out}}$ be a sequence of high-quality (HQ) target frames. $T$, $H$, $W$, $C_{in}$ and $C_{out}$ are the frame number, height, width and input channel number and output channel number, respectively. $s$ is the upscaling factor, which is larger than 1 (*e.g*., for video SR) or equal to 1 (*e.g*., for video deblurring). The proposed Video Restoration

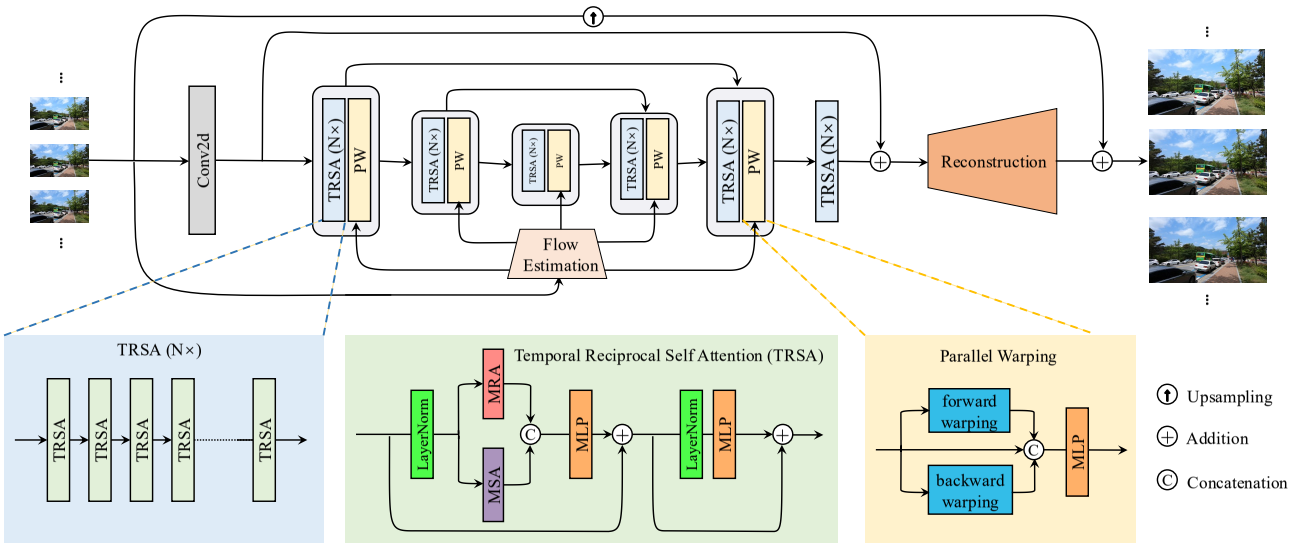



Figure 4.2: The framework of the proposed Video Restoration Transformer (VRT). Given $T$ low-quality input frames, VRT reconstructs $T$ high-quality frames in parallel. It jointly extracts features, deals with misalignment, and fuses temporal information at multiple scales. On each scale, it has two kinds of modules: temporal reciprocal self attention (TRSA, see Sec. 4.3.2) and parallel warping (see Sec. 4.3.3). The downsampling and upsampling operations between different scales are omitted for clarity.

Transformer (VRT) aims to restore $T$ HQ frames from $T$ LQ frames in parallel for various video restoration tasks, including video SR, deblurring, denoising, *etc*. As illustrated in Fig. 4.2, VRT can be divided into two parts: feature extraction and reconstruction.

FEATURE EXTRACTION. At the beginning, we extract shallow features $I^{SF} \in \mathbb{R}^{T\times H\times W\times C}$ by a single spatial 2D convolution from the LQ sequence $I^{LQ}$. After that, based on [232], we propose a multi-scale network that aligns frames at different image resolutions. More specifically, when the total scale number is $S$, we downsample the feature for $S-1$ times by squeezing each $2\times 2$ neighborhood to the channel dimension and reducing the channel number to the original number via a linear layer. Then, we upsample the feature gradually by unsqueezing the feature back to its original size. In such a way, we can extract features and deal with object or camera motions at different scales by two kinds of modules: temporal reciprocal self attention (TRSA, see 4.3.2) and parallel warping (see 4.3.3). Skip connections are added for features of same scales. Finally, after multi-scale feature extraction, alignment and fusion, we add several TRSA modules for further feature refinement and obtain the deep feature $I^{DF} \in \mathbb{R}^{T\times H\times W\times C}$.

RECONSTRUCTION. After feature extraction, we reconstruct the HQ frames from the addition of shallow feature $I^{SF}$ and deep feature $I^{DF}$. Different frames are reconstructed independently based on their corresponding features. Besides, to ease the burden of feature learning, we employ global residual learning and only predict the residual between the bilinearly upsampled LQ sequence and the ground-truth HQ sequence. In practice, different reconstruction modules are used for different restoration tasks. For video SR, we use the sub-pixel convolution layer [141] to upsample the feature by a scale factor of $s$. For video deblurring, a single convolution layer is enough for reconstruction. Apart from this, the architecture designs are kept the same for all tasks.

LOSS FUNCTION. For fair comparison with existing methods, we use the commonly used Charbonnier loss [145] between the reconstructed HQ sequence $I^{RHQ}$ and the ground-truth HQ sequence $I^{HQ}$ as

$$\mathcal{L} = \sqrt{\|I^{RHQ} - I^{HQ}\|^2 + \epsilon^2}, \tag{4.1}$$

where $\epsilon$ is a constant that is empirically set as $10^{-3}$.

### 4.3.2 *Temporal Reciprocal Self Attention*

In this section, based on the attention mechanism [99], [225], [226], we first introduce the reciprocal attention and then propose the temporal reciprocal self attention (TRSA).

RECIPROCAL ATTENTION. Given a reference frame feature $X^R \in \mathbb{R}^{N\times C}$ and a supporting frame feature $X^S \in \mathbb{R}^{N\times C}$, where $N$ is the number of feature elements and $C$ is the channel number, we compute the *query* $Q^R$, *key* $K^S$ and *value* $V^S$ from $X^R$ and $X^S$ by linear projections as

$$Q^R = X^R P^Q, \quad K^S = X^S P^K, \quad V^S = X^S P^V, \tag{4.2}$$

where $P^Q, P^K, P^V \in \mathbb{R}^{C\times D}$ are projection matrices. $D$ is the channel number of projected features. Then, we use $Q^R$ to query $K^S$ in order to generate the attention map $A = \text{SoftMax}(Q^R(K^S)^T/\sqrt{D}) \in \mathbb{R}^{N\times N}$, which is then used for weighted sum of $V^S$. This is formulated as

$$\text{MA}(Q^R, K^S, V^S) = \text{SoftMax}(Q^R(K^S)^T/\sqrt{D})V^S, \tag{4.3}$$

where SoftMax means the row softmax operation.

Since $Q^R$ and $K^S$ come from $X^R$ and $X^S$, respectively, $A$ reflects the correlation between elements in the reference image and the supporting image. For clarity, we rewrite Eq. (4.3) for the $i$-th element of the reference image as

$$Y^R_{i,:} = \sum_{j=1}^{N} A_{i,j} V^S_{j,:}, \tag{4.4}$$

where $Y^R_{i,:}$ refers to the new feature of the $i$-th element in the reference frame. As shown in Fig. 4.3, when $K^S_{k,:}$ (*e.g.*, the yellow square from the supporting frame) is the most similar element to $Q^R_{i,:}$ (*e.g.*, the orange square from the reference frame), $A_{i,k} > A_{i,j}$ holds for all $j \neq k$ ($j \leq N$). When all $K^S_{j,:}(j \neq k)$ are very dissimilar to $Q^R_i$, we have

$$\begin{cases} A_{i,k} \to 1, \\ A_{i,j} \to 0, \qquad\qquad for \;\; j \neq k, j \leq N. \end{cases} \tag{4.5}$$

In this extreme case, by combining Eq. (4.4) and (4.5), we have $Y^R_{i,:} = V^S_{k,:}$, which moves the $k$-th element in the supporting frame to the position of the $i$-th element in the reference frame (see the dashed red

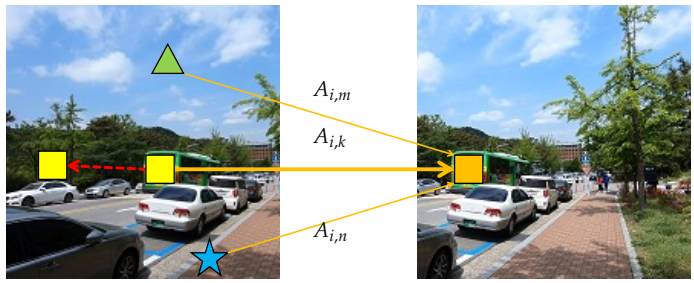



Figure 4.3: Illustration for reciprocal attention. We let the orange square (the $i$-th element of the reference frame) query elements in the supporting frame and use their weighted features as a new representation for the orange square. The weights are shown around solid arrows (we only show three examples for clarity). When $A_{i,k} \to 1$ and the rest $A_{i,j} \to 0 (j \neq k)$, the reciprocal attention equals to warping the yellow square to the position of the orange square (illustrated as a dashed arrow).

line in Fig. 4.3). This equals to image warping given an optical flow vector. When $A_{i,k} \to 1$ does not hold, Eq. (4.4) can be regarded as a "soft" version of image warping. In practice, the reference frame and supporting frame can be exchanged, allowing mutual alignment between two frames. Besides, similar to multi-head self attention, we can also perform the attention for $h$ times and concatenate the results as multi-head reciprocal attention (MRA).

Particularly, reciprocal attention has several benefits over the combination of explicit motion estimation and image warping. First, reciprocal attention can adaptively preserve information from the supporting frame than image warping, which only focuses on the target pixel. It also avoids black hole artifacts when there is no matched positions. Second, reciprocal attention does not have the inductive biases of locality, which is inherent to most CNN-based motion estimation methods [220], [233]–[235] and may lead to performance drop when two neighboring objects move towards different directions. Third, reciprocal attention equals to conducting motion estimation and warping on image features in a joint way. In contrast, optical flows are often estimated on the input

RGB image and then used for warping on features [176], [177]. Besides, flow estimation on RGB images is often not robust to lighting variation, occlusion and blur [215].

TEMPORAL RECIPROCAL SELF ATTENTION (TRSA). Reciprocal attention is proposed for joint feature alignment between two frames. To extract and preserve feature from the current frame, we use reciprocal attention together with self attention. Let $X \in \mathbb{R}^{2\times N\times C}$ represent two frames, which can be split into $X_1 \in \mathbb{R}^{1\times N\times C}$ and $X_2 \in \mathbb{R}^{1\times N\times C}$. We use multi-head reciprocal attention (MRA) on $X_1$ and $X_2$ for two times: warping $X_1$ towards $X_2$ and warping $X_2$ towards $X_1$. The warped features are combined and then concatenated with the result of multi-head self attention (MSA), followed by a multi-layer perceptron (MLP) for the purpose of dimension reduction. After that, another MLP is added for further feature transformation. Two LayerNorm (LN) layers and two residual connections are also used as shown in the green box of Fig. 4.2. The whole process formulated as follows

$$\begin{aligned} X_1, X_2 &= \text{Split}_0(\text{LN}(X)) \\ Y_1, Y_2 &= \text{MRA}(X_1, X_2), \text{MRA}(X_2, X_1) \\ Y_3 &= \text{MSA}(Concat_0(X_1, X_2)) \\ X &= \text{MLP}(\text{Concat}_2(\text{Concat}_0(Y_1, Y_2), Y_3)) + X \\ X &= \text{MLP}(\text{LN}(X)) + X \end{aligned} \tag{4.6}$$

where the subscripts of Split and Concat refer to the specified dimensions. However, due to the design of reciprocal attention, Eq. (4.6) can only deal with two frames at a time.

One naive way to extend Eq. (4.6) for $T$ frames is to deal with frame-to-frame pairs exhaustively, resulting in the computational complexity of $\mathcal{O}(T^2)$. Inspired by the shifted window mechanism [103], [231], we propose the temporal reciprocal self attention (TRSA) to remedy the problem. TRSA first partitions the video sequence into non-overlapping 2-frame clips and then applies Eq. (4.6) to them in parallel. Next, as shown in Fig. 4.4, it shifts the sequence temporally by 1 frame for every other layer to enable cross-clip connections, reducing the computational complexity to $\mathcal{O}(T)$. The temporal receptive field size is increased when multiple TRSA modules are stacked together. Specifically, at layer $i$ ($i \geq 2$), one frame can utilize information from up to $2(i-1)$ frames.

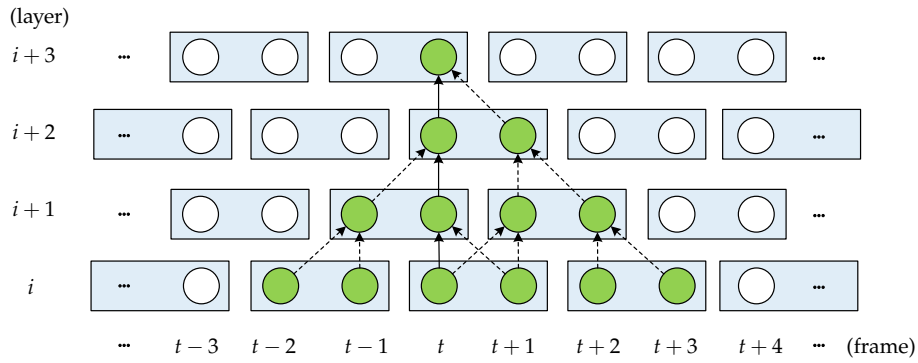



Figure 4.4: Illustration for temporal reciprocal self attention (TRSA). It shows a stack of temporal reciprocal self attention (TRSA) layers. The sequence is partitioned into 2-frame clips at each layer and shifted for every other layer to enable cross-clip interactions. Dashed lines represent information fusion among different frames.

DISCUSSION. Video restoration tasks often need to process high-resolution frames. Since the complexity of attention is quadratic to the number of elements within the attention window, global attention on the full image is often impractical. Therefore, following [1], [103], we partition each frame spatially into non-overlapping $M \times M$ local windows, resulting in $\frac{HW}{M^2}$ windows. Shifted window mechanism (with the shift of $\lfloor \frac{M}{2} \rfloor \times \lfloor \frac{M}{2} \rfloor$ pixels) is also used spatially to enable cross-window connections. Besides, although stacking multiple TRSA modules allows for long-distance temporal modelling, distant frames are not directly connected. As will show in the ablation study, using only a small temporal window size cannot fully exploit the potential of the model. Therefore, we use larger temporal window size for the last quarter of TRSA modules to enable direct interactions between distant frames.

### 4.3.3 *Parallel Warping*

Due to spatial window partitioning, the reciprocal attention mechanism may not be able to deal with large motions well. Hence, as shown in the orange box of Fig. 4.2, we use feature warping at the end of each network stage to handle large motions. As shown in Fig. 4.5, for any frame feature $X_t$, we calculate the optical flows of its neighbouring

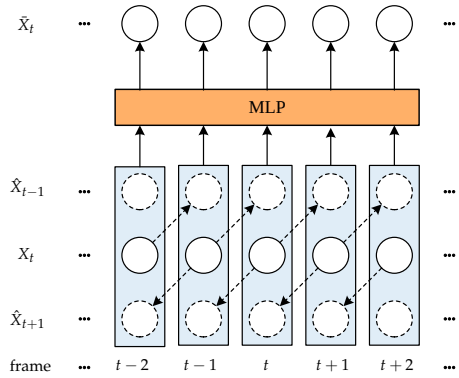



Figure 4.5: Illustration of parallel warping. For every frame feature $X_t (t \leq T)$, frame $X_{t-1}$ and $X_{t+1}$ are warped towards $X_t$ as $\hat{X}_{t-1}$ and $\hat{X}_{t+1}$, respectively. Then, $X_t$, $\hat{X}_{t-1}$ and $\hat{X}_{t+1}$ are concatenated together (denoted by blue boxes) for feature fusion and dimension reduction with a multi-layer perception (MLP). The final output is $\bar{X}_t$. The dashed arrows and circles denote warping operations and warped features, respectively.

frame features $X_{t-1}$ and $X_{t+1}$, and warp them towards the frame $X_t$ as $\hat{X}_{t-1}$ and $\hat{X}_{t+1}$ (*i.e.*, backward and forward warping). Then, we concatenate $X_t$, $\hat{X}_{t-1}$ and $\hat{X}_{t+1}$ along the channel dimension (denoted by the blue box). To keep the original channel size for later operations, we reduce its dimension by a multi-layer perception (MLP) and obtain $\bar{X}_t$. This mechanism can be generalized for four (*i.e.*, $X_{t-2}$, $X_{t-1}$, $X_{t+1}$ and $X_{t+2}$) and six (*i.e.*, $X_{t-3}$, $X_{t-2}$, $X_{t-1}$, $X_{t+1}$, $X_{t+2}$ and $X_{t+3}$) neighboring frames. Note that different frames are processed in parallel.

Specifically, following [177], we predict the residual flow by a flow estimation model and use deformable convolution [223] for deformable alignment. Given estimated optical flows $O_{t-1,t}$ and $O_{t+1,t}$, we first use them to warp $X_{t-1}$ and $X_{t+1}$, respectively as

$$\begin{cases} X'_{t-1} = \mathcal{W}(X_{t-1}, O_{t-1,t}), \\ X'_{t+1} = \mathcal{W}(X_{t+1}, O_{t+1,t}), \end{cases} \tag{4.7}$$

where $\mathcal{W}$ represents the image warping function. $X'_{t-1}$ and $X'_{t+1}$ are the initial warped features. Then, we use several convolution layers (denoted as $\mathcal{C}$) to predict the offset residuals $o_{t-1,t}$, $o_{t+1,t}$ and modulation

masks $m_{t-1,t}, m_{t+1,t}$ from the concatenation of $O_{t-1,t}$, $O_{t+1,t}$, $X'_{t-1}$ and $X'_{t+1}$ as

$$o_{t-1,t}, o_{t+1,t}, m_{t-1,t}, m_{t+1,t} = \mathcal{C}(Concat(O_{t-1,t}, O_{t+1,t}, X'_{t-1}, X'_{t+1})). \quad (4.8)$$

Next, we warp $X_{t-1}$ and $X_{t+1}$ again as

$$\begin{cases} \hat{X}_{t-1} = \mathcal{D}(X_{t-1}, O_{t-1,t} + o_{t-1,t}, m_{t-1,t}), \\ \hat{X}_{t+1} = \mathcal{D}(X_{t+1}, O_{t+1,t} + o_{t+1,t}, m_{t+1,t}), \end{cases} \quad (4.9)$$

where $\mathcal{D}$ refers to the deformable convolution. The outputs $\hat{X}_{t-1}$ and $\hat{X}_{t+1}$ and concatenated with $X_t$ as the new feature for the $t$-th frame.

## 4.4 EXPERIMENTS

### 4.4.1 *Experimental Setup*

ARCHITECTURE. For video SR, we use 4 scales for VRT. On each scale, we stack 8 TRSA modules, the last two of which use a temporal window size of 8. The spatial window size $M \times M$, head size $h$, and channel size $C$ are set to $8 \times 8$, 6 and 120, respectively. After 7 multi-scale feature extraction stages, we add 24 TRSA modules (only with self attention) for further feature extraction before reconstruction. For flow estimation, we extract multi-scale flows from different layers of SpyNet [220], [234] and feed them into different scales of VRT. In video SR, for the additional TRSA modules in the 8-th stage, the channel number is set as 180. We use temporal window sizes of 8 and 2 for the first two thirds of modules and the rest ones, respectively. In video deblurring and denoising, we use a relatively smaller model. The channel sizes for the first 7 stages and the 8-th stage are 96 and 120, respectively. For the 8-th stage, we only use 16 TRSA modules. In addition, the gated variant GEGLU [224] is used to replace the plain feed-forward network.

TRAINING. The training batch size and total number of iteration are 8 and 300K, respectively. We augment the input frames by random flipping, rotation and cropping. The model is trained by the Adam optimizer [148] with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. The learning rate is initialized as $4e-4$ and decreased gradually according to the Cosine Annealing scheme [236]. To stabilize training, we use the pretrained

model of SpyNet for initialization and fix the flow estimation part for the first 20K iterations. We also use a smaller initial learning rate (*i.e.*, $5e-5$) for it. There are several training differences in different tasks. First, we set the training frame number as 6 for most tasks (7 for Vimeo-90K in video SR) and additionally provide experiments on 16 frames for REDS in video SR. For Vimeo-90K, following [177], we initialize the model with the REDS pretrained model. Second, the training patch size is $64 \times 64$ for video SR and $192 \times 192$ for other tasks. Third, for video denoising, we follow [25], [26] and train a non-blind denoising model using varying noise levels ($\sigma \sim \mathcal{U}(0, 50)$) by concatenating the noise level map with the noisy video along the channel dimension. All experiments are conducted on a server with 8 A100 GPUs. For video SR, it takes about 5 and 10 days for 6-frame and 16-frame experiments, respectively. For video deblurring and denoising, the training time is about 10 days.

DATASET. For video super-resolution, we train the model on two different training datasets for scale factor 4. First, we generate low-resolution images by the MATLAB `imresize` function (*i.e.*, bicubic degradation) and train the model on REDS [237]. REDS4 [24] is used as the test set. Second, we train the model on Vimeo-90K [215] with two different degradations: bicubic and blur downsampling (Gaussian blur with $\sigma = 1.6$ followed by subsampling). The testing datasets include Vimeo-90K-T [215], Vid4 [238] and UDM10 [207]. For video deblurring, we train the model on three different datasets (DVD [180], GoPro [21] and REDS [237]). We test it on their corresponding testing sets (for REDS, we use REDS4 [24]). For video denoising, we train the model on the DAVIS [239] and test it on the corresponding testing set and Set8 [25]. The details of datasets are as follows.

1. REDS [237]. REDS is a newly-proposed high-quality ($1280 \times 720$) video dataset for video restoration. It has 270 clips for training and validation. Following [24], we use REDS4 (4 selected representative clips, *i.e.*, 000, 011, 015 and 020) for evaluation and the rest 266 clips for training. This dataset is used for training bicubic video SR.

2. Vimeo-90K [215]. Vimeo-90K is a widely-used middle-quality ($448 \times 256$) dataset for video restoration. For video SR benchmarking, it uses 64,612 clips for training and 7,824 clips for testing

(denoted as Vimeo-90K-T). This dataset is used for training bicubic and blur-downsampling video SR.

3. Vid4 [238]. Vid4 is a classical testing dataset for video restoration. It contains 4 video clips (*i.e.*, calendar, city, foliage and walk). Each clip has at least 34 frames ($720 \times 480$).

4. UDM10 [207]. UDM10 is a recent proposed testing dataset for video super-resolution. It contains 4 video clips of various scenes, each of which has 32 frames ($1272 \times 720$).

5. DVD [180]. DVD is a widely-used high-quality ($1280 \times 720$) dataset for video deblurring. Blurred images are generated from high fps videos. It has 61 videos (5,708 frames in total) for training and 10 videos (1,000 frames in total) for testing.

6. GoPro [21]. GoPro is a popular high-quality ($1280 \times 720$) for image and video deblurring. Similar to DVD [180], blurred images are synthesized based on high fps videos. It is consisted of 22 training clips (2,103 frames in total) and 11 testing clips (1,111 frames in total).

7. DAVIS [239]. DAVIS-2017 is a popular middle-quality ($854 \times 480$) dataset for video denoising. It consists of 90 videos for training and 30 videos for testing.

8. Set8 [26]. Set8 consists of 8 middle quality ($960 \times 540$) videos (*i.e.*, tractor, touchdown, park_joy, sunflower, hypersmooth, motorbike, rafting and snowboard). It is often used as a testing dataset in video denoising. Following [25], [26], [204], we only use the first 85 frames of each video.

EVALUATION. For evaluation, following [24], [25], [28], [176], [183], we calculate the metrics on RGB channel for REDS4 [24], DVD testing set [180], GoPro testing set [21], DAVIS testing set [239] as well as Set8 [25], and on the Y channel for Vimeo-90K-T [215], Vid4 [238] and UDM10 [207].

### 4.4.2 *Video Super-Resolution*

#### 4.4.2.1 *Quantitative Results*

As shown in Table 4.1, we compare VRT with the state-of-the-art image and video SR methods [8], [24], [171], [173], [175]–[177], [182], [187], [189], [190], [202], [207], [215], [221], [240]. VRT achieves best performance for both bicubic (BI) and blur-downsamplng (BD) degradations. Specifically, when trained on the REDS [237] dataset with short sequences, VRT outperforms VSRT by up to 0.57dB in PSNR. Compared with another representative sliding window-based model EDVR, VRT has an improvement of 0.50∼1.57dB on different datasets, showing its good ability to fuse information from multiple frames. Note that VRT outputs all frames simultaneously rather than predicting them frame by frame as EDVR does. On the Vimeo-90K [215] dataset, VRT surpasses BasicVSR++ by up to 0.38dB, although BasicVSR++ and other recurrent models may mirror the 7-frame video for training and testing. When VRT is trained on longer sequences, it shows good potential in temporal modelling and further increases the PSNR by 0.52dB. As indicated in [8], recurrent models often suffer from significant performance drops on short sequences. In contrast, VRT performs well on both short and long sequences. We note that VRT is slightly lower than the 32-frame model BasicVSR++. This is expected as VRT is trained on 16 frames.

We also provide comparison on parameter number and runtime in Table 4.1. As a parallel model, VRT needs to restore all frames at the same time, which leads to relatively larger model size and longer runtime per frame compared with recurrent models. However, VRT has the potential for distributed deployment, which is hard for recurrent models that restore a video clip recursively by design.

#### 4.4.2.2 *Qualitative Results*

Visual results of different methods are shown in Fig. 4.6. As one can see, in accordance with its significant quantitative improvements, VRT can generate visually pleasing images with sharp edges and fine details, such as horizontal strip patterns of buildings. By contrast, its competitors suffer from either distorted textures or lost details.

Table 4.1: Quantitative comparison (average PSNR/SSIM) with state-of-the-art methods for video super-resolution (×4) on REDS4 [237], Vimeo-90K-T [215], Vid4 [238] and UDM10 [207]. Best and second best results are in red and blue colors, respectively. †We currently do not have enough GPU memory to train the fully parallel model VRT on 30 frames.

| Method | Training Frames (REDS/ Vimeo-90K) | Params (M) | Runtime (ms) | BI degradation | | | BD degradation | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | REDS4 [237] (RGB channel) | Vimeo-90K-T [215] (Y channel) | Vid4 [238] (Y channel) | UDM10 [207] (Y channel) | Vimeo-90K-T [215] (Y channel) | Vid4 [238] (Y channel) |
| Bicubic | - | - | - | 26.14/0.7292 | 31.32/0.8684 | 23.78/0.6347 | 28.47/0.8253 | 31.30/0.8687 | 21.80/0.5246 |
| SwinIR [1] | - | 11.9 | - | 29.05/0.8269 | 35.67/0.9287 | 25.68/0.7491 | 35.42/0.9380 | 34.12/0.9167 | 25.25/0.7262 |
| SwinIR-ft [1] | 1/1 | 11.9 | - | 29.24/0.8319 | 35.89/0.9301 | 25.69/0.7488 | 36.76/0.9467 | 35.70/0.9293 | 25.62/0.7498 |
| TOFlow [215] | 5/7 | - | - | 27.98/0.7990 | 33.08/0.9054 | 25.89/0.7651 | 36.26/0.9438 | 34.62/0.9212 | 25.85/0.7659 |
| FRVSR [187] | 10/7 | 5.1 | 137 | - | - | - | 37.09/0.9522 | 35.64/0.9319 | 26.69/0.8103 |
| DUF [221] | 7/7 | 5.8 | 974 | 28.63/0.8251 | - | 27.33/0.8319 | 38.48/0.9605 | 36.87/0.9447 | 27.38/0.8329 |
| PFNL [207] | 7/7 | 3.0 | 295 | 29.63/0.8502 | 36.14/0.9363 | 26.73/0.8029 | 38.74/0.9627 | - | 27.16/0.8355 |
| RBPN [188] | 7/7 | 12.2 | 1507 | 30.09/0.8590 | 37.07/0.9435 | 27.12/0.8180 | 38.66/0.9596 | 37.20/0.9458 | 27.17/0.8205 |
| MuCAN [171] | 5/7 | - | - | 30.88/0.8750 | 37.32/0.9465 | - | - | - | - |
| RLSP [173] | -/7 | 4.2 | 49 | - | - | - | 38.48/0.9606 | 36.49/0.9403 | 27.48/0.8388 |
| TGA [182] | -/7 | 5.8 | 384 | - | - | - | 38.74/0.9627 | 37.59/0.9516 | 27.63/0.8423 |
| RSDN [175] | -/7 | 6.2 | 94 | - | - | - | 39.35/0.9653 | 37.23/0.9471 | 27.92/0.8505 |
| RRN [189] | -/7 | 3.4 | 45 | - | - | - | 38.96/0.9644 | - | 27.69/0.8488 |
| FDAN [190] | -/7 | 9.0 | - | - | - | - | 39.91/0.9686 | 37.75/0.9522 | 27.88/0.8508 |
| EDVR [24] | 5/7 | 20.6 | 378 | 31.09/0.8800 | 37.61/0.9489 | 27.35/0.8264 | 39.89/0.9686 | 37.81/0.9523 | 27.85/0.8503 |
| GOVSR [202] | -/7 | 7.1 | 81 | - | - | - | 40.14/0.9713 | 37.63/0.9503 | 28.41/0.8724 |
| VSRT [8] | 5/7 | 32.6 | - | 31.19/0.8815 | 37.71/0.9494 | 27.36/0.8258 | - | - | - |
| **VRT (ours)** | 6/- | 30.7 | 236 | 31.60/0.8888 | - | - | - | - | - |
| BasicVSR [176] | 15/14 | 6.3 | 63 | 31.42/0.8909 | 37.18/0.9450 | 27.24/0.8251 | 39.96/0.9694 | 37.53/0.9498 | 27.96/0.8553 |
| IconVSR [176] | 15/14 | 8.7 | 70 | 31.67/0.8948 | 37.47/0.9476 | 27.39/0.8279 | 40.03/0.9694 | 37.84/0.9524 | 28.04/0.8570 |
| BasicVSR++[177] | 30/14 | 7.3 | 77 | 32.39/0.9069† | 37.79/0.9500 | 27.79/0.8400 | 40.72/0.9722 | 38.21/0.9550 | 29.04/0.8753 |
| **VRT (ours)** | 16/7 | 35.6 | 243 | 32.19/0.9006 | 38.20/0.9530 | 27.93/0.8425 | 41.05/0.9737 | 38.72/0.9584 | 29.42/0.8795 |

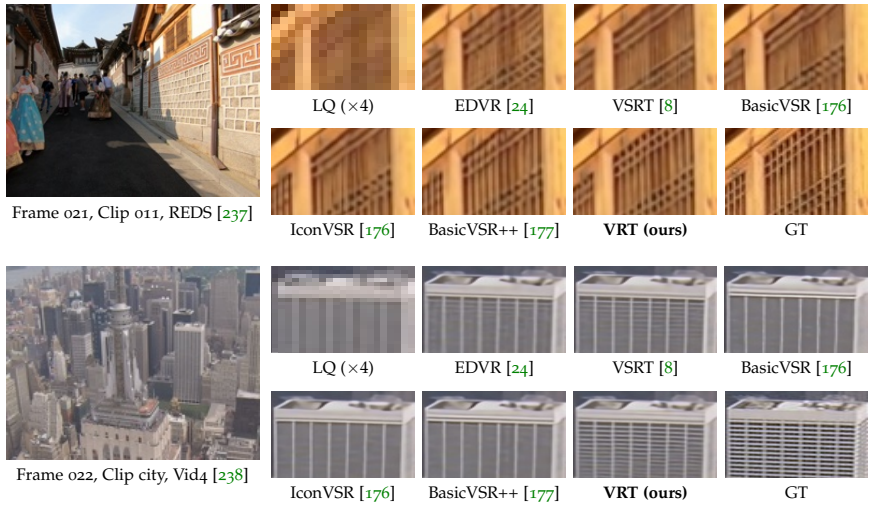



Figure 4.6: Visual comparison of video super-resolution (×4) methods.

Table 4.2: Video SR (×4, BI degradation) results on Vimeo-Fast/ Medium/ Slow subsets.

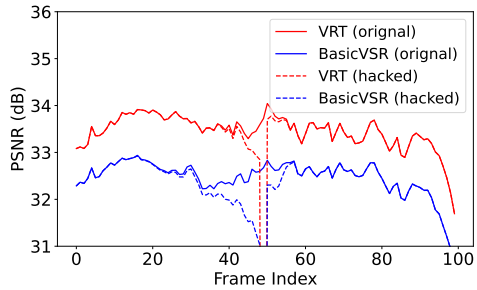
| Subset | EDVR [24] | BasicVSR [176] | BasicVSR++ [177] | **VRT (ours)** |
|---|---|---|---|---|
| Fast | 40.77 | 40.34 | 40.98 | 41.44 |
| Medium | 37.81 | 37.35 | 37.99 | 38.42 |
| Slow | 34.52 | 34.11 | 34.57 | 34.98 |

#### 4.4.2.3 *Performance in Different Motion Conditions*

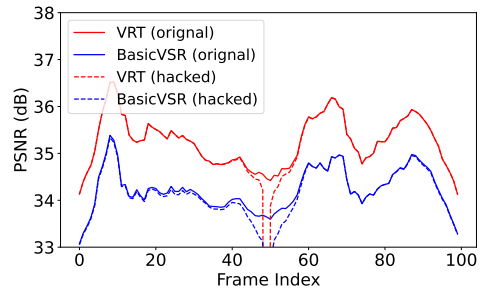
Following [209], we compare different methods on Vimeo90K [215] (×4, BI degradation) with fast/ medium/ slow motions. As shown in Table 4.2, VRT leads to larger improvement on fast motion videos than on slow ones when compared with existing methods.

#### 4.4.2.4 *Robustness to Noise*

In addition, to compare the noise robustness of parallel models and recurrent models, we hack the LQ input video by manually setting all pixels of the 50-th frame as zero in testing. As shown in Fig. 4.7, VRT suffers from less performance drop and has less adverse impact on neighbouring frames than BasicVSR, indicating that VRT is more robust to noise.




(a) Clip 011, REDS [237]




(b) Clip 015, REDS [237]

Figure 4.7: The robustness to noise injection attack. It compares the per-frame PSNR drop of different methods, when pixels of the 50-th frame of the LQ input video is hacked to be all zeros during testing.

#### 4.4.2.5 *Attention Visualization*

To show exploit what the attention mechanism has learned, we plot attention maps between a pixel (marked as red points) from the first frame and the rest pixels in the same attention window. As shown in Fig. 4.8, the first row is the input image patch, while the rest rows are the visualization of attention matrices of six different attention heads. The attention matrices are normalized for visualization, whereas brighter pixels means larger attention weights. As shown in Fig. 4.8a, when the red point moves towards the top-right direction from the first frame to the last frame, it moves most attention to the top-right direction as well. Similar observations can be concluded from other examples. This shows that, in the attention mechanism, one pixel is able to find its most related pixels and attend to it, bringing long-range dependency modelling ability of our model across different frames.

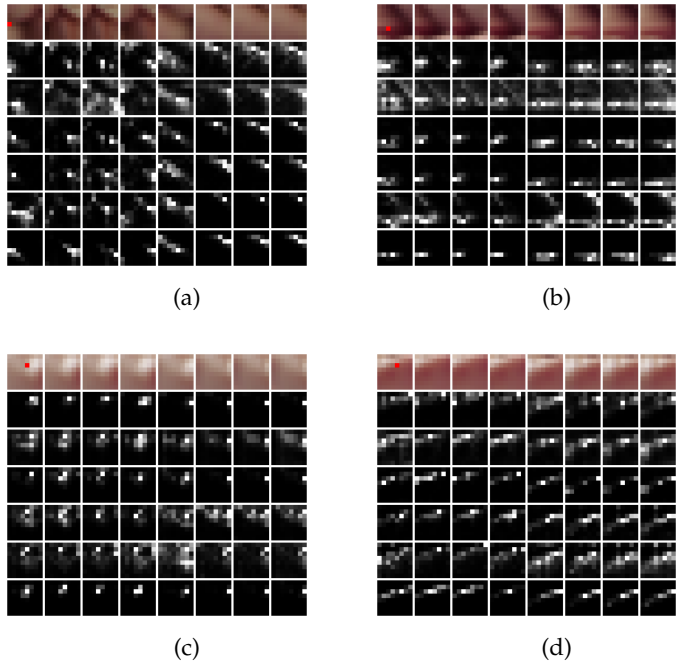



Figure 4.8: Visualization of attention maps. The first row shows the original image patches at the same position from different frames, while the rest rows are the attention weight visualizations of six different attention heads. The query pixel is marked by a red point in the first frame.

### 4.4.3 *Video Deblurring*

#### 4.4.3.1 *Quantitative Results*

We conducts experiments on three different datasets for fair comparison with existing methods [21], [24], [28], [180], [181], [183], [184], [186], [208], [210], [241], [242]. Table 4.3 shows the results on the DVD [180] dataset. It is clear that VRT achieves the best performance, outperforming the second best method ARVo by a remarkable improvement of 1.47dB and 0.0299 in terms of PSNR and SSIM. PVDNet proposes motion estimation learning to better aggregate information from multiple frames, but it is inferior to the proposed VRT, which uses reciprocal attention for alignment. Related to the attention mechanism, GSTA designs a gated spatio-temporal attention mechanism, while ARVo calculates the correlation between pixel pairs for correspondence learn-

Table 4.3: Quantitative comparison (average RGB channel PSNR/SSIM) with state-of-the-art methods for video deblurring on DVD [180]. Following [183], [210], all restored frames instead of randomly selected 30 frames from each test set [180] are used in evaluation.

| Method | DeepDeblur [21] | SRN [241] | DBLRNet [242] | STFAN [181] | STTN [222] | SFE [208] |
|---|---|---|---|---|---|---|
| PSNR | 29.85 | 30.53 | 30.08 | 31.24 | 31.61 | 31.71 |
| SSIM | 0.8800 | 0.8940 | 0.8845 | 0.9340 | 0.9160 | 0.9160 |
| Method | EDVR [24] | TSP [210] | PVDNet [186] | GSTA [28] | ARVo [183] | **VRT (ours)** |
| PSNR | 31.82 | 32.13 | 32.31 | 32.53 | 32.80 | 34.27 (+1.47) |
| SSIM | 0.9160 | 0.9268 | 0.9260 | 0.9468 | 0.9352 | 0.9651 (+0.03) |

Table 4.4: Quantitative comparison (average RGB channel PSNR/SSIM) with state-of-the-art methods for video deblurring on GoPro [21].

| Method | DeepDeblur [21] | SRN [241] | SAPHN [243] | MPRNet [244] | SFE [208] | IFI-RNN [184] |
|---|---|---|---|---|---|---|
| PSNR | 29.23 | 30.26 | 31.85 | 32.66 | 31.01 | 31.05 |
| SSIM | 0.9162 | 0.9342 | 0.9480 | 0.9590 | 0.9130 | 0.9110 |
| Method | ESTRNN [185] | EDVR [24] | TSP [210] | PVDNet [186] | GSTA [28] | **VRT (ours)** |
| PSNR | 31.07 | 31.54 | 31.67 | 31.98 | 32.10 | 34.81 (+2.15) |
| SSIM | 0.9023 | 0.9260 | 0.9279 | 0.9280 | 0.9600 | 0.9724 (+0.01) |

ing. However, both of them are based on CNN, achieving significantly worse performance compared with the Transformer-based VRT. We also compare VRT on the GoPro [21] (Table 4.4) and REDS [237] (Table 4.5) datasets. VRT shows its superiority over other methods with significant PSNR gains of 2.15dB and 1.99dB. The total number of parameters of VRT is 18.3M, which is slightly smaller than EDVR (23.6M) and PVDNet (23.5M). The runtime is 2.2s per frame on $1280 \times 720$ blurred videos. Notably, during evaluation, we do not use any pre-processing techniques such as sequence truncation and image alignment [186], [210].

#### 4.4.3.2 *Qualitative Results*

Fig. 4.9 shows the visual comparison of different methods. VRT is effective in removing motion blurs and restoring faithful details, such

Table 4.5: Quantitative comparison (average RGB channel PSNR/SSIM) with state-of-the-art methods for video deblurring on REDS [237].

| Method | DeepDeblur [21] | SRN [241] | DBN [180] | EDVR [24] | **VRT (ours)** |
|---|---|---|---|---|---|
| PSNR | 26.16 | 26.98 | 26.55 | 34.80 | 36.79 (+1.99) |
| SSIM | 0.8249 | 0.8141 | 0.8066 | 0.9487 | 0.9648 (+0.02) |

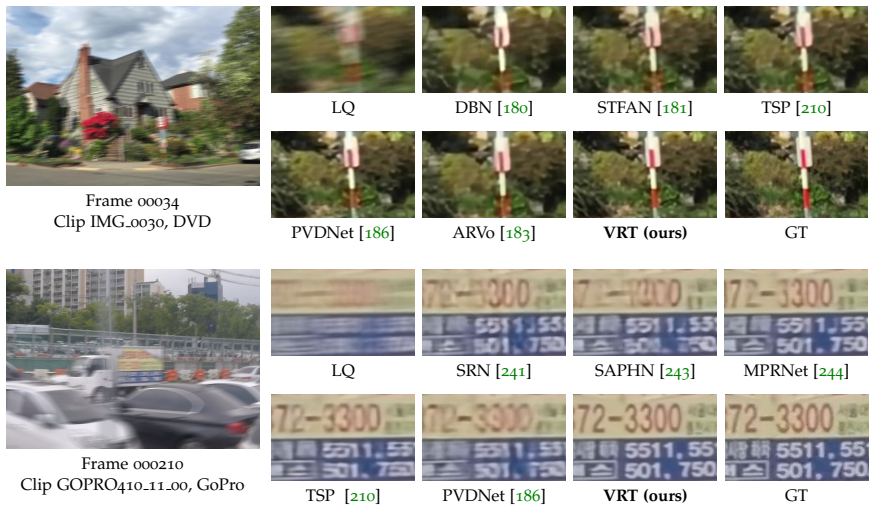



Figure 4.9: Visual comparison of video deblurring methods. Part of compared images are derived from [183], [244].

as the pole in the first example and characters in the second one. In comparison, other approaches fail to remove blurs completely and do not produce sharp edges.

In addition, we conduct a user study with 20 users on video deblurring. Each user is given multiple pairs of deblurred videos from DVD [180], where one is our result. As shown in Fig. 4.10, over 90% of the users vote that VRT has better visual quality than existing methods.

### 4.4.4 *Video Denoising*

We also conduct experiments on video denoising to show the effectiveness of VRT. Following [25], [26], we train one non-blind model for noise level $\sigma \in [0, 50]$ on the DAVIS [239] dataset and test it on

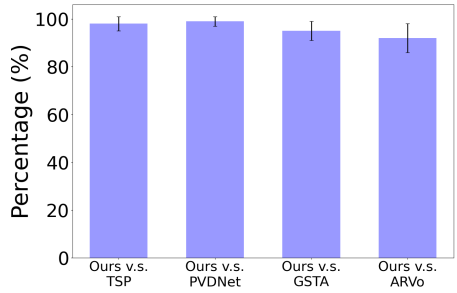



Figure 4.10: User study of video deblurring on the DVD [180] dataset.

different noise levels. Table 4.6 shows the superiority of VRT on two benchmark datasets over existing methods [25], [26], [204], [245]. Even though PaCNet [204] trains different models separately for different noise levels, VRT still improves the PSNR by 0.82∼2.16dB.

### 4.4.5 *Video Frame Interpolation*

To show the generalizability of our framework, we conduct experiments on video frame interpolation. Following [246], [247], we train the model on Vimeo-90K [215] for single frame interpolation and test it on quintuples generated from Vimeo-90K-T [215], UCF101 [248] and DAVIS [239]. As shown in Table 4.7, VRT achieves best or competitive performance on all datasets compared with it competitors, including those using depth maps or optical flows. As for the model size, VRT only has 9.9M parameters, which is much smaller than the recent best model FLAVR (42.4M).

### 4.4.6 *Space-Time Video Super-Resolution*

With the pretrained models on video SR (VSR) and video frame interpolation (VFI), we directly test VRT on space-time video super-resolution by cascading VRT models in two ways: VFI followed by VSR, or VSR followed by VFI. As shown in Table 4.8, compared with existing methods, VRT provides a strong baseline for space-time video super-resolution, even though it serves as a two-stage model and is not specifically

Table 4.6: Quantitative comparison (average RGB channel PSNR) with state-of-the-art methods for video denoising on DAVIS [239] and Set8 [25]. $\sigma$ is the additive white Gaussian noise level.

| Dataset | $\sigma$ | VLNB [245] | DVDnet [25] | FastDVDnet [26] | PaCNet [204] | **VRT (ours)** |
|---|---|---|---|---|---|---|
| DAVIS | 10 | 38.85 | 38.13 | 38.71 | 39.97 | 40.82 (+0.85) |
| | 20 | 35.68 | 35.70 | 35.77 | 36.82 | 38.15 (+1.33) |
| | 30 | 33.73 | 34.08 | 34.04 | 34.79 | 36.52 (+1.73) |
| | 40 | 32.32 | 32.86 | 32.82 | 33.34 | 35.32 (+1.98) |
| | 50 | 31.13 | 31.85 | 31.86 | 32.20 | 34.36 (+2.16) |
| Set8 | 10 | 37.26 | 36.08 | 36.44 | 37.06 | 37.88 (+0.82) |
| | 20 | 33.72 | 33.49 | 33.43 | 33.94 | 35.02 (+1.08) |
| | 30 | 31.74 | 31.79 | 31.68 | 32.05 | 33.35 (+1.30) |
| | 40 | 30.39 | 30.55 | 30.46 | 30.70 | 32.15 (+1.45) |
| | 50 | 29.24 | 29.56 | 29.53 | 29.66 | 31.22 (+1.56) |

Table 4.7: Quantitative comparison (average RGB channel PSNR) with state-of-the-art methods for video frame interpolation (single frame interpolation, $\times 2$) on Vimeo-90K-T [215], UCF101 [248] and DAVIS [239]. R, D and F means that the model uses RGB images, depth maps or optical flows.

| Method | Inputs | Vimeo-90K-T [215] | UCF101 [248] | DAVIS [239] |
|---|---|---|---|---|
| DAIN [249] | R+D+F | 33.35/0.945 | 31.64/0.957 | 26.12/0.870 |
| QVI [246] | R+F | 35.15/0.971 | 32.89/0.970 | 27.17/0.874 |
| DVF [250] | R | 27.27/0.893 | 28.72/0.937 | 22.13/0.800 |
| SepConv [30] | R | 33.60/0.944 | 31.97/0.943 | 26.21/0.857 |
| CAIN [251] | R | 33.93/0.964 | 32.28/0.965 | 26.46/0.856 |
| SuperSloMo [29] | R | 32.90/0.957 | 32.33/0.960 | 25.65/0.857 |
| BMBC [252] | R | 34.76/0.965 | 32.61/0.955 | 26.42/0.868 |
| AdaCoF [253] | R | 35.40/0.971 | 32.71/0.969 | 26.49/0.866 |
| FLAVR [247] | R | 36.25/0.975 | 33.31/0.971 | 27.43/0.874 |
| **VRT (ours)** | R | 36.53/0.977 | 33.30/0.970 | 27.88/0.889 |

trained for this task. In particular, it improves the PSNR by 1.03dB on the Vid4 dataset.

### 4.4.7 *Ablation Study*

For ablation study, we set up a small version of VRT as the baseline model by halving the layer and channel numbers. All models are trained on Vimeo-90K [215] for bicubic video SR ($\times 4$) and tested it on Vid4 [238].

Table 4.8: Quantitative comparison (average Y channel PSNR) with state-of-the-art methods for space-time video super-resolution (time: $\times 2$, space: $\times 4$) on Vid4 [238] and Vimeo-90K-T [215]. [29], [30] and [249] are frame interpolation methods SuperSloMo, SepConv and DAIN, respectively. Note that the proposed VRT is not trained on this task. We directly test it by cascading pre-trained video super-resolution (VSR) and video frame interpolation (VFI) models.

| VFI+VSR Methods | Vid4 [238] | Vimeo-Fast [215] | Vimeo-Medium [215] | Vimeo-Slow [215] |
|---|---|---|---|---|
| [29]+Bicubic | 22.84/0.5772 | 31.88/0.8793 | 29.94/0.8477 | 28.37/0.8102 |
| [29]+RCAN [18] | 23.80/0.6397 | 34.52/0.9076 | 32.50/0.8884 | 30.69/0.8624 |
| [29]+RBPN [188] | 23.76/0.6362 | 34.73/0.9108 | 32.79/0.8930 | 30.48/0.8584 |
| [29]+EDVR [24] | 24.40/0.6706 | 35.05/0.9136 | 33.85/0.8967 | 30.99/0.8673 |
| [30]+Bicubic | 23.51/0.6273 | 32.27/0.8890 | 30.61/0.8633 | 29.04/0.8290 |
| [30]+RCAN [18] | 24.92/0.7236 | 34.97/0.9195 | 33.59/0.9125 | 32.13/0.8967 |
| [30]+RBPN [188] | 26.08/0.7751 | 35.07/0.9238 | 34.09/0.9229 | 32.77/0.9090 |
| [30]+EDVR [24] | 25.93/0.7792 | 35.23/0.9252 | 34.22/0.9240 | 32.96/0.9112 |
| [249]+Bicubic | 23.55/0.6268 | 32.41/0.8910 | 30.67/0.8636 | 29.06/0.8289 |
| [249]+RCAN [18] | 25.03/0.7261 | 35.27/0.9242 | 33.82/0.9146 | 32.26/0.8974 |
| [249]+RBPN [188] | 25.96/0.7784 | 35.55/0.9300 | 34.45/0.9262 | 32.92/0.9097 |
| [249]+EDVR [24] | 26.12/0.7836 | 35.81/0.9323 | 34.66/0.9281 | 33.11/0.9119 |
| ZSM [209] | 26.31/0.7976 | 36.81/0.9415 | 35.41/0.9361 | 33.36/0.9138 |
| STARnet [254] | 26.06/0.8046 | 36.19/0.9368 | 34.86/0.9356 | 33.10/0.9164 |
| TMNet [255] | 26.43/0.8016 | 37.04/0.9435 | 35.60/0.9380 | 33.51/0.9159 |
| RSTT [256] | 26.43/0.7994 | 36.80/0.9403 | 35.66/0.9381 | 33.50/0.9147 |
| VRT (VFI+VSR) | 26.59/0.8014 | 36.56/0.9372 | 35.28/0.9343 | 33.75/0.9204 |
| VRT (VSR+VFI) | 27.46/0.8392 | 36.98/0.9439 | 36.01/0.9434 | 34.01/0.9236 |

#### 4.4.7.1 *Impact of multi-scale architecture and parallel warping.*

Table 4.9 shows the ablation study on the multi-scale architecture and parallel warping. When the number of model scales is reduced, the performance drops gradually, even though the computation burden becomes heavier. This is expected because multi-scale processing can help the model utilize information from a larger area and deal with large motions between frames. Besides, parallel warping also helps, bringing an improvement of 0.17dB.

#### 4.4.7.2 *Impact of temporal reciprocal self attention.*

To test the effectiveness of reciprocal and self attention in TRSA, we conduct ablation study in Table 4.10. When we replace reciprocal

Table 4.9: Ablation study on multi-scale architecture and parallel warping. Given an input of spatial size $64 \times 64$, the corresponding feature sizes of each scale are shown in brackets. When some scales are removed, we add more layers to the rest scales for similar model size.

| 1 ($64 \times 64$) | 2 ($32 \times 32$) | 3 ($16 \times 16$) | 4 ($8 \times 8$) | Parallel Warping | PSNR |
|---|---|---|---|---|---|
| ✓ | | | | ✓ | 27.13 |
| ✓ | ✓ | | | ✓ | 27.20 |
| ✓ | ✓ | ✓ | | ✓ | 27.25 |
| ✓ | ✓ | ✓ | ✓ | | 27.11 |
| ✓ | ✓ | ✓ | ✓ | ✓ | 27.28 |

Table 4.10: Ablation study on temporal reciprocal self attention.

| Attention 1 | Self Attn. | - | Reciprocal Attn. | Reciprocal Attn. |
|---|---|---|---|---|
| Attention 2 | Self Attn. | Self Attn. | - | Self Attn. |
| PSNR | 27.17 | 27.11 | 26.92 | 27.28 |

attention with self attention (*i.e.*, two self attentions) or only use one self attention, the performance drops by 0.11∼0.17dB. One possible reason is that the model may be more focused on the reference frame rather than on the supporting frame during the computation of attention maps. In contrast, using the reciprocal attention can help the model to explicitly attend to the supporting frame and benefit from feature fusion. In addition, we can find that only using reciprocal attention is not enough. This is because reciprocal attention cannot preserve information of reference frames.

#### 4.4.7.3 *Impact of attention window size.*

We conduct ablation study in Table 4.11 to investigate the impact of attention window size in the last few TRSAs of each scale. When the temporal window size increases from 1 to 2, the performance only improves slightly, possibly due to the fact that previous TRSA layers can already make good use of neighboring two-frame information. When the size is increased to 8, we can see an obvious improvement of 0.18dB. As a result, we use the window size of $8 \times 8 \times 8$ for those layers.

Table 4.11: Ablation study on attention window size (frame × height × width).

| Window Size | $1 \times 8 \times 8$ | $2 \times 8 \times 8$ | $4 \times 8 \times 8$ | $8 \times 8 \times 8$ |
|---|---|---|---|---|
| PSNR | 27.10 | 27.13 | 27.18 | 27.28 |

## 4.5 CONCLUSION

In this chapter, we proposed the Video Restoration Transformer (VRT) for video restoration. Based on a multi-scale framework, it jointly extracts, aligns, and fuses information from different frames at multiple resolutions by two kinds of modules: multiple temporal reciprocal self attention (TRSA) and parallel warping. More specifically, TRSA is composed of reciprocal and self attention. Reciprocal attention allows joint implicit flow estimation and feature warping, while self attention is responsible for feature extraction. Parallel warping is also used to further enhance feature alignment and fusion. Extensive experiments on various benchmark datasets show that VRT brings significant performance gains (up to 2.16dB) for various video restoration tasks, including video super-resolution, video deblurring and video denoising.

# 5 RECURRENT VIDEO RESTORATION TRANSFORMER

Existing video restoration methods generally fall into two extreme cases, *i.e.*, they either restore all frames in parallel (such as the Video Restoration Transformer proposed in Chapter 4) or restore the video frame by frame in a recurrent way, which would result in different merits and drawbacks. Typically, the former has the advantage of temporal information fusion. However, it suffers from large model size and intensive memory consumption; the latter has a relatively small model size as it shares parameters across frames; however, it lacks long-range dependency modeling ability and parallelizability.

This chapter attempt to integrate the advantages of the two cases by proposing a recurrent video restoration transformer, namely RVRT. RVRT processes local neighboring frames in parallel within a globally recurrent framework which can achieve a good trade-off between model size, effectiveness, and efficiency. Specifically, RVRT divides the video into multiple clips and uses the previously inferred clip feature to estimate the subsequent clip feature. Within each clip, different frame features are jointly updated with implicit feature aggregation. Across different clips, the guided deformable attention is designed for clip-to-clip alignment, which predicts multiple relevant locations from the whole inferred clip and aggregates their features by the attention mechanism. Extensive experiments on video super-resolution, deblurring, and denoising show that the proposed RVRT achieves state-of-the-art performance on benchmark datasets with balanced model size, testing memory and runtime.

## 5.1 INTRODUCTION

Video restoration, such as video super-resolution, deblurring, and denoising, has become a hot topic in recent years. It aims to restore a clear and sharp high-quality video from a degraded (*e.g.*, downsampled, blurred, or noisy) low-quality video [2], [8], [24], [177]. It has wide applications in live streaming [257], video surveillance [258], old film restoration [259], and more.

Parallel methods and recurrent methods have been dominant strategies for solving various video restoration problems. Typically, those two kinds of methods have their respective merits and demerits. Parallel methods [2], [8], [24], [171], [172], [178]–[183] support distributed deployment and achieve good performance by directly fusing information from multiple frames, but they often have a large model size and consume enormous memory for long-sequence videos. In the meanwhile, recurrent models [173]–[177], [184]–[187], [189], [190], [240] reuse the same network block to save parameters and predict the new frame feature based on the previously refined frame feature, but the sequential processing strategy inevitably leads to information loss and noise amplification [260] for long-range dependency modelling and makes it hard to be parallelized.

Considering the advantages and disadvantages of parallel and recurrent methods, in this chapter, we propose a recurrent video restoration transformer (RVRT) that takes the best of both worlds. On the one hand, RVRT introduces the recurrent design into transformer-based models to reduce model parameters and memory usage. On the other hand, it processes neighboring frames together as a clip to reduce video sequence length and alleviate information loss. To be specific, we first divide the video into fixed-length video clips. Then, starting from the first clip, we refine the subsequent clip feature based on the previously inferred clip feature and the old features of the current clip from shallower layers. Within each clip, different frame features are jointly extracted, implicitly aligned and effectively fused by the self-attention mechanism [1], [99], [103]. Across different clips, information is accumulated clip by clip with a larger hidden state than previous recurrent methods.

To implement the above RVRT model, one big challenge is how to align different video clips when using the previous clip for feature refinement. Most existing alignment techniques [2], [24], [172], [176], [177], [187], [215], [220], [223], [235] are designed for frame-to-frame alignment. One possible way to apply them to clip-to-clip alignment is by introducing an extra feature fusion stage after aligning all frame pairs. Instead, we propose an one-stage video-to-video alignment method named guided deformable attention (GDA). More specifically, for a reference location in the target clip, we first estimate the coordinates of multiple relevant locations from different frames in the supporting clip under the guidance of optical flow, and then aggregate features of all locations dynamically by the attention mechanism.

GDA has several advantages over previous alignment methods: 1) Compared with optical flow-based warping that only samples one point from one frame [176], [187], [215], GDA benefits from multiple relevant locations sampled from the video clip. 2) Unlike mutual attention [2], GDA utilizes features from arbitrary locations without suffering from the small receptive field in local attention or the huge computation burden in global attention. Besides, GDA allows direct attention on non-integer locations with bilinear interpolation. 3) In contrast to deformable convolution [24], [172], [177], [216], [223], [261] that uses a fixed weight in feature aggregation, GDA generates dynamic weights to aggregate features from different locations. It also supports arbitrary location numbers and allows for both frame-to-frame and video-to-video alignment without any modification.

Our contributions can be summarized as follows:

- We propose the recurrent video restoration transformer (RVRT) that extracts features of local neighboring frames from one clip in a joint and parallel way, and refines clip features by accumulating information from previous clips and previous layers. By reducing the video sequence length and transmitting information with a larger hidden state, RVRT alleviates information loss and noise amplification in recurrent networks, and also makes it possible to partially parallelize the model.
- We propose the guided deformable attention (GDA) for one-stage video clip-to-clip alignment. It dynamically aggregates information of relevant locations from the supporting clip.
- Extensive experiments on eight benchmark datasets show that the proposed model achieves state-of-the-art performance in three challenging video restoration tasks: video super-resolution, video deblurring, and video denoising, with balanced model size, memory usage and runtime.

## 5.2 RELATED WORK

### 5.2.1 *Video Restoration*

PARALLEL VS. RECURRENT METHODS. Most existing video restoration methods can be classified as parallel or recurrent methods ac-

cording to their parallelizability. Parallel methods estimate all frames simultaneously, as the refinement of one frame feature is not dependent on the update of other frame features. They can be further divided as sliding window-based methods [24]–[26], [171], [172], [178]–[183], [211], [212] and transformer-based methods [2], [8]. The former kind of methods typically restore merely the center frame from the neighboring frames and are often tested in a sliding window fashion rather than in parallel. These methods generally consist of four stages: feature extraction, feature alignment, feature fusion, and frame reconstruction. Particularly, in the feature alignment stage, they often align all frames towards the center frame, which leads to quadratic complexity with respect to video length and is hard to be extended for long-sequence videos. Instead, the latter kind of method reconstructs all frames at a time based on the transformer architectures. They jointly extract, align, and fuse features for all frames, achieving significant performance improvements against previous methods. However, current transformer-based methods are laid up with a huge model size and large memory consumption. Different from above parallel methods, recurrent methods [10], [173]–[177], [184]–[187], [189], [190], [209], [240], [262] propagate latent features from one frame to the next frame sequentially, where information of previous frames is accumulated for the restoration of later frames. Basically, they are composed of three stages: feature extraction, feature propagation and frame reconstruction. The features are propagated from the first to the last frame in a recurrent way. Due to the recurrent nature of feature propagation, recurrent methods suffer from information loss and the inapplicability of distributed deployment. However, they often have a lightweight model design thanks to the reuse and refinement of differnt frame features.

ALIGNMENT IN VIDEO RESTORATION. Unlike image restoration that mainly focuses on feature extraction [4]–[7], [12], [17]–[19], [50], [198], how to align multiple highly-related but misaligned frames is another key problem in video restoration. Traditionally, many methods [176], [178], [214], [215], [217]–[219], [263] first estimate the optical flow between neighbouring frames [220], [233], [235] and then conduct image warping for alignment. Other techniques, such as deformable convolution [8], [24], [172], [177], [223], [261], dynamic filter [221] and mutual attention [2], have also been exploited for implicit feature alignment.

### 5.2.2 *Vision Transformer*

Transformer [99] is the de-facto standard architecture in natural language processing. Recently, it has been used in dealing with vision problems by viewing pixels or image patches as tokens [100], [102], achieving remarkable performance gains in various computer vision tasks, including image classification [102], [103], [132], [264], object detection [106], [134], [265], semantic segmentation [131], [138], [266], *etc*. It also achieves promising results in restoration tasks [1], [2], [8], [13], [104], [139], [256], [262], [267]–[271]. In particular, for video restoration, Cao *et al*. [8] propose the first transformer model for video SR, while Liang *et al*. [2] propose an unified framework for video SR, deblurring and denoising.

We note that some transformer-based works [265], [272] have tried to combine the concept of deformation [223], [261] with the attention mechanism [99]. Zhu *et al*. [272] directly predicts the attention weight from the query feature without considering its feature interaction with supporting locations. Xia *et al*. [265] place the supporting points uniformly on the image to make use of global information. Both above two methods are proposed for recognition tasks such as object detection, which is fundamentally different from video alignment in video restoration. Lin *et al*. [262] use pixel-level or patch-level attention to aggregate information from neighbouring frames under the guidance of optical flow, but it only samples one supporting pixel or patch from one frame, restricting the model from attending to multiple distant locations.

## 5.3 METHODOLOGY

### 5.3.1 *Overall Architecture*

Given a low-quality video sequence $I^{LQ} \in \mathbb{R}^{T\times H\times W\times C}$, where $T$, $H$, $W$ and $C$ are the video length, height, width and channel, respectively, the goal of video restoration is to reconstruct the high-quality video $I^{HQ} \in \mathbb{R}^{T\times sH\times sW\times C}$, where $s$ is the scale factor. To reach this goal, we propose a recurrent video restoration transformer, as illustrated in Fig. 5.1. The model consists of three parts: shallow feature extraction, recurrent feature refinement and HQ frame reconstruction. More specifically, in shallow feature extraction, we first use a convolution layer to extract features from the LQ video. For deblurring and

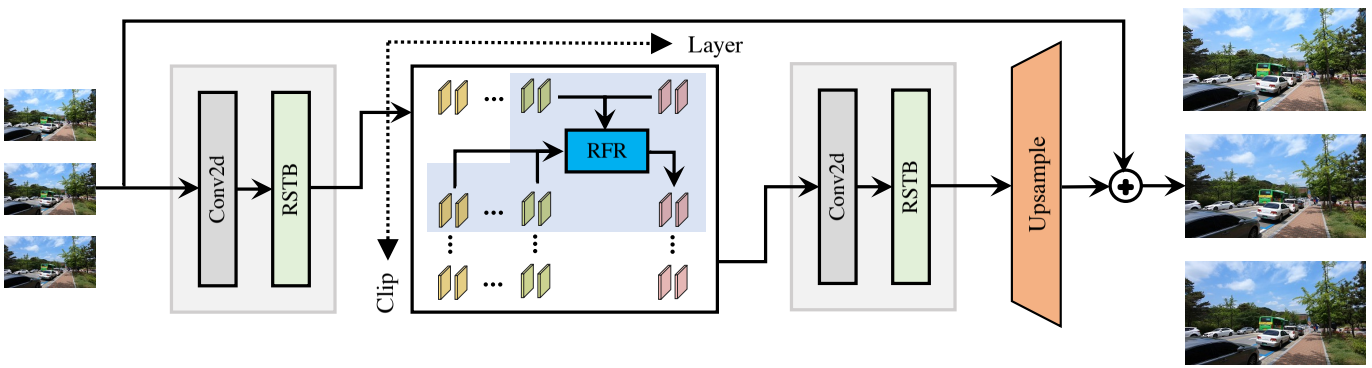



Figure 5.1: The architecture of recurrent video restoration transformer (RVRT). From left to right, it consists of shallow feature extraction, recurrent feature refinement and HQ frame reconstruction. In recurrent feature refinement (RFR, see more details in Fig. 5.2), we divide the video into $N$-frame clips ($N = 2$ in this figure) and process frames in one clip in parallel within a globally recurrent framework in time. Multiple refinement layers are stacked for better performance.

denoising (*i.e.*, $s = 1$), we additionally add two strided convolution layers to downsample the feature and reduce computation burden in the next layers. After that, several Residual Swin Transformer Blocks (RSTBs) [1] are used to extract the shallow feature. Then, we use recurrent feature refinement modules for temporal correspondence modeling and guided deformable attention for video alignment, which are detailed in Sec. 5.3.2 and Sec. 5.3.3, respectively. Lastly, we add several RSTBs to generate the final feature and reconstruct the HQ video $I^{RHQ}$ by pixel shuffle layer [141]. For training, the Charbonnier loss [145] $\mathcal{L} = \sqrt{\|I^{RHQ} - I^{HQ}\|^2 + \epsilon^2}$ ($\epsilon = 10^{-3}$) is used for all tasks.

### 5.3.2 *Recurrent Feature Refinement*

We stack $L$ recurrent feature refinement modules to refine the video feature by exploiting the temporal correspondence between different frames. To make a trade-off between recurrent and transformer-based methods, we process $N$ frames locally in parallel on the basis of a globally recurrent framework.

Formally, given the video feature $F^i \in \mathbb{R}^{T\times H\times W\times C}$ from the $i$-th layer, we first reshape it as a 5-dimensional tensor $F^i \in \mathbb{R}^{\frac{T}{N}\times N\times H\times W\times C}$ by dividing it into $\frac{T}{N}$ video clip features: $F^i_1, F^i_2, ..., F^i_{\frac{T}{N}} \in \mathbb{R}^{N\times H\times W\times C}$. Each clip feature $F^i_t$ ($1 \leq t \leq \frac{T}{N}$) has $N$ neighbouring frame features: $F^i_{t,1}, F^i_{t,2}, ..., F^i_{t,N} \in \mathbb{R}^{H\times W\times C}$. To utilize information from neighbouring clips, we align the $(t-1)$-th clip feature $F^i_{t-1}$ towards the $t$-th clip based on the optical flow $O^i_{t-1\to t}$, clip feature $F^{i-1}_{t-1}$ and clip feature $F^{i-1}_t$. This is formulated as follows:

$$\widehat{F}^i_{t-1} = GDA(F^i_{t-1}; O^i_{t-1\to t}, F^{i-1}_{t-1}, F^{i-1}_t), \tag{5.1}$$

where $GDA$ is the guided deformable attention and $\widehat{F}^i_{t-1}$ is the aligned clip feature. The details of GDA will be described in Sec. 5.3.3.

Similar to recurrent neural networks [176], [177], [187], as shown in Fig. 5.2, we update the clip feature of each time step as follows:

$$F^i_t = RFR(F^0_t, F^1_t, ..., F^{i-1}_t, \widehat{F}^i_{t-1}), \tag{5.2}$$

where $F^0_t$ is the output of the shallow feature extraction module and $F^1_t, F^2_t, ..., F^{i-1}_t$ are from previous recurrent feature refinement modules. $RFR(\cdot)$ is the recurrent feature refinement module that consists of a

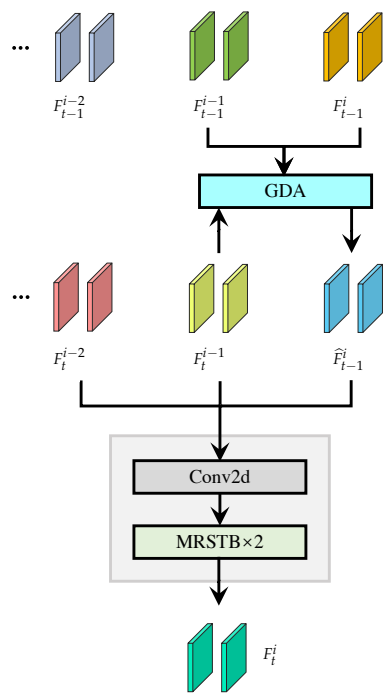



Figure 5.2: The illustrations of recurrent feature refinement (RFR). The $(t-1)$-th clip feature $F_{t-1}^{i}$ from the $i$-th layer is aligned towards the $t$-th clip as $\widehat{F}_{t-1}^{i}$ by guided deformable attention (GDA, see more details in Fig. 5.3). $F_{t}^{0}, F_{t}^{1}, ..., F_{t}^{i-1}$ and $\widehat{F}_{t-1}^{i}$ are then refined as $F_{t}^{i}$ by several modified residual swin transformer blocks (MRSTBs), in which different frames are jointly processed in a parallel way.

convolution layer for feature fusion and several modified residual Swin Transformer blocks (*MRSTBs*) for feature refinement. In *MRSTB*, we upgrade the original 2D $h \times w$ attention window to the 3D $N \times h \times w$ attention window, so that every frame in the clip can attend to itself and other frames simultaneously, allowing implicit feature aggregation. In addition, in order to accumulate information forward and backward in time, we reverse the video sequence for all even recurrent feature refinement modules [174], [177].

The above recurrent feature refinement module is the key component of the proposed RVRT model. Globally, features of different video clips are propagated in a recurrent way. Locally, features of different frames are updated jointly in parallel. For an arbitrary single frame, it

can make full use of global information accumulated in time and local information extracted together by the self-attention mechanism. As we can see, RVRT is a generalization of both recurrent and transformer models. It becomes a recurrent model when $N = 1$ or a transformer model when $N = T$. This is fundamentally different from previous methods that adopt transformer blocks to replace CNN blocks within a recurrent architecture [259], [262]. It is also different from existing attempts in natural language processing [273], [274].

### 5.3.3 *Guided Deformable Attention for Video Alignment*

Different from previous frameworks, the proposed RVRT needs to align neighboring related but misaligned video clips, as indicated in Eq. (5.1). In this subsection, we propose the guided deformation attention (GDA) for video clip-to-clip alignment.

Given the $(t-1)$-th clip feature $F^i_{t-1}$ from the $i$-th layer, our goal is to align $F^i_{t-1}$ towards the $t$-th clip as a list of features $\widehat{F}^{i,(1:N)}_{t-1} = \widehat{F}^{i,(1)}_{t-1}, \widehat{F}^{i,(2)}_{t-1}, ..., \widehat{F}^{i,(N)}_{t-1}$, where $\widehat{F}^{i,(n)}_{t-1} (1 \leq n \leq N)$ denotes the aligned clip feature towards the $n$-th frame feature $F^i_{t,n}$ of the $t$-th clip, and $\widehat{F}^{i,(n)}_{t-1,n'} (1 \leq n' \leq N)$ is the aligned frame feature from the $n'$-th frame in the $(t-1)$-th clip to the $n$-th frame in the $t$-th clip. Inspired by optical flow estimation designs [177], [233]–[235], [262], we first pre-align $F^{i,(n)}_{t-1,n'}$ with the optical flow $O^{i,(n)}_{t-1\to t,n'}$ as $\bar{F}^{i,(n)}_{t-1,n'} = \mathcal{W}(F^{i,(n)}_{t-1,n'}, O^{i,(n)}_{t-1\to t,n'})$, where $\mathcal{W}$ denotes the warping operation. For convenience, we summarize the pre-alignments of all "$n'$-to-$n$" $(1 \leq n', n \leq N)$ frame pairs between the $(t-1)$-th and $t$-th video clips as follows:

$$\bar{F}^{i,(1:N)}_{t-1} = \mathcal{W}(F^i_{t-1}, O^{i,(1:N)}_{t-1\to t}), \tag{5.3}$$

After that, we predict the optical flow offsets $o^{i,(1:N)}_{t-1\to t}$ from the concatenation of $F^{i-1}_t$, $\bar{F}^{i(1:N)}_{t-1}$ and $O^{i,(1:N)}_{t-1\to t}$ along the channel dimension. A small convolutional neural network (CNN) with several convolutional layers and ReLU layers is used for prediction. This is formulated as

$$o^{i,(1:N)}_{t-1\to t} = CNN(Concat(F^{i-1}_t, \bar{F}^{i,(1:N)}_{t-1}, O^{i,(1:N)}_{t-1\to t})), \tag{5.4}$$

where the current misalignment between the $t$-th clip feature and the warped $(t-1)$-th clip features can reflect the offset required for

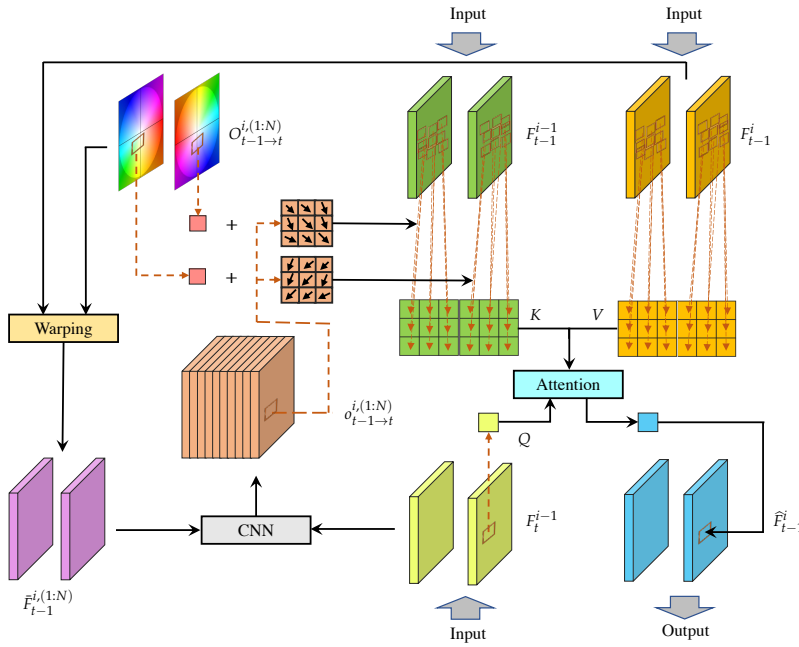



Figure 5.3: The illustrations of guided deformable attention (GDA). We estimate offsets of multiple relevant locations from different frames based on the warped clip, and then aggregate features of different locations dynamically by the attention mechanism. $F^i_{t-1}$ is the $(t-1)$-th clip feature from the $i$-th layer, while $\bar{F}^i_{t-1}$ and $\hat{F}^i_{t-1}$ are the pre-aligned and aligned features of $F^i_{t-1}$. $O^{i,(1:N)}_{t-1\to t}$ and $o^{i,(1:N)}_{t-1\to t}$ denote optical flows and offsets, respectively.

further alignment. In practice, we initialize $O^{1,(1:N)}_{t-1\to t}$ as the optical flows estimated from the LQ input video via SpyNet [220], and predict $M$ offsets for each frame ($NM$ offsets in total). The optical flows are updated layer by layer as follows:

$$O^{i+1,(n)}_{t-1\to t,n'} = O^{i,(n)}_{t-1\to t,n'} + \frac{1}{M}\sum_{m=1}^{M}\{o^{i,(n)}_{t-1\to t,n'}\}_m, \tag{5.5}$$

where $\{o^{i,(n)}_{t-1\to t,n'}\}_m$ denotes the $m$-th offset in $M$ predictions from the $n'$-th frame to the $n$-th frame.

Then, for the $n$-th frame of the $t$-th clip, we sample its relevant features from the $(t-1)$-th clip feature $F^i_{t-1}$ according the predicted locations, which are indicated by the sum of optical flow and offsets,

*i.e.*, $O^{i,(n)}_{t-1\to t} + o^{i,(n)}_{t-1\to t}$, according to the chain relationship $F^{i}_{t-1} \xrightarrow{O^{i,(n)}_{t-1\to t}} \bar{F}^{i,(n)}_{t-1} \xrightarrow{o^{i,(n)}_{t-1\to t}} \widehat{F}^{i,(n)}_{t-1}$ [177], [220]. For simplicity, we define the queries $Q$, keys $K$ and values $V$ as follows:

$$Q = F^{i-1}_{t,n} P_Q, \tag{5.6}$$

$$K = Sampling(F^{i-1}_{t-1} P_K, O^{i,(n)}_{t-1\to t} + o^{i,(n)}_{t-1\to t}), \tag{5.7}$$

$$V = Sampling(F^{i}_{t-1} P_V, O^{i,(n)}_{t-1\to t} + o^{i,(n)}_{t-1\to t}), \tag{5.8}$$

where $Q \in \mathbb{R}^{1\times C}$ is the projected feature from the $n$-th frame of $t$-th clip. $K \in \mathbb{R}^{NM\times C}$ and $V \in \mathbb{R}^{NM\times C}$ are the projected features that are bilinearly sampled from $NM$ locations of $F^{i-1}_{t-1}$ and $F^{i}_{t-1}$, respectively. $P_Q \in \mathbb{R}^{C\times C}$, $P_K \in \mathbb{R}^{C\times C}$ and $P_V \in \mathbb{R}^{C\times C}$ are the projection matrices. Note that we first project the feature and then do sampling to reduce redundant computation.

Next, similar to the attention mechanism [99], we calculate the attention weights based on the $Q$ and $K$ from the $(i-1)$-th layer and then compute the aligned feature $\widehat{F}^{i,(n)}_{t-1}$ as a weighted sum of $V$ from the same $i$-th layer as follows:

$$\widehat{F}^{i,(n)}_{t-1} = SoftMax(QK^T/\sqrt{C}))V, \tag{5.9}$$

where *SoftMax* is the softmax operation along the row direction and $\sqrt{C}$ is a scaling factor.

Lastly, since Eq. (5.9) only aggregates information spatially, we add a multi-layer perception (*MLP*) with two fully-connected layers and a *GELU* activation function between them to enable channel interaction as follows:

$$\widehat{F}^{i}_{t-1} = \widehat{F}^{i}_{t-1} + MLP(\widehat{F}^{i}_{t-1}), \tag{5.10}$$

where a residual connection is used to stabilize training. The hidden and output channel numbers of the *MLP* are $RC$ ($R$ is the ratio) and $C$, respectively.

MULTI-GROUP MULTI-HEAD GUIDED DEFORMABLE ATTENTION. We can divide the channel into several deformable groups and perform the deformable sampling for different groups in parallel. Besides, in the attention mechanism, we can further divide one deformable group into several attention heads and perform the attention operation separately for different heads. All groups and heads are concatenated together before channel interaction.

CONNECTION TO DEFORMABLE CONVOLUTION. Deformable convolution [223], [261] uses a learned weight for feature aggregation, which can be seen as a special case of GDA, *i.e.*, using different projection matrix $P_V$ for different locations and then directly averaging the resulting features. Its parameter number and computation complexity are $MC^2$ and $\mathcal{O}(MC^2)$, respectively. In contrast, GDA uses the same projection matrix for all locations but generates dynamic weights to aggregate them. Its parameter number and computation complexity are $(3+2R)C^2$ and $\mathcal{O}((3C+2RC+M)C)$, which are similar to deformable convolution when choosing proper $M$ and $R$.

## 5.4 EXPERIMENTS

### 5.4.1 *Experimental Setup*

ARCHITECTURE. For shallow feature extraction and HQ frame reconstruction, we use 1 RSTB that has 2 swin transformer layers. For recurrent feature refinement, we use 4 refinement modules with a clip size of 2, each of which has 2 MRSTBs with 2 modified swin transformer layers. For both RSTB and MRSTB, spatial attention window size and head number are $8 \times 8$ and 6, respectively. We use 144 channels for video SR and 192 channels for deblurring and denoising. In GDA, we use 12 deformable groups and 12 deformable heads with 9 candidate locations. We empirically project the query to a higher-dimensional space (*e.g.*, 2C) because we found it can improve the performance slightly and the parameter number of GDA is not a bottleneck.

TRAINING. In training, we randomly crop $256 \times 256$ HQ patches and use different video lengths for different datasets: 30 frames for REDS [237], 14 frames for Vimeo-90K [215], and 16 frames for DVD [180], GoPro [21] as well as DAVIS [239]. Adam optimizer [148] with default setting is used to train the model for 600,000 iterations when the batch size is 8. The learning rate is initialized as $4 \times 10^{-4}$ and deceased with the Cosine Annealing scheme [236]. To stabilize training, we initialize SpyNet [220], [234] with pretrained weights, fix it for the first 30,000 iterations and reduce its learning rate by 75%. For video super-resolution, we train the model on two different training datasets for scale factor 4. First, we generate low-resolution images by the MATLAB `imresize` function (*i.e.*, bicubic degradation) and train the model on

REDS [237]. REDS4 [24] (*i.e.*, clip 000, 011, 015, 020) is used as the test set. Second, we train the model on Vimeo-90K [215] with two different degradations: bicubic and blur downsampling (Gaussian blur with $\sigma = 1.6$ followed by subsampling). The testing datasets include Vimeo-90K-T [215], Vid4 [238] and UDM10 [207]. On 8 Nvidia A100 GPUs, it takes about 17 days. For video deblurring, we train the model on two different datasets DVD [180] and GoPro [21]. The training time is about 10 days. We test it on their corresponding testing sets. For video denoising, we train the model on the DAVIS [239] and test it on the corresponding testing set and Set8 [25]. The training time is similar to deblurring. We train all models on 8 Nvidia A100 GPUs. It takes about 16.6 days for video SR and 9.7 days for video deblurring and denoising. For training memory cost, it consumes about 39GB and 29GB for video SR and other two tasks, respectively.

EVALUATION. Following [24], [25], [28], [176], [183], we calculate the metrics on RGB channel for REDS4 [24], DVD testing set [180], GoPro testing set [21], DAVIS testing set [239] as well as Set8 [25], and on the Y channel for Vimeo-90K-T [215], Vid4 [238] and UDM10 [207].

### 5.4.2 *Video Super-Resolution*

For video SR, we consider two settings: bicubic (BI) and blur-downsampling (BD) degradation. For BI degradation, we train the model on two different datasets: REDS [237] and Vimeo-90K [215], and then test the model on their corresponding testsets: REDS4 and Vimeo-90K-T. We additionally test Vid4 [238] along with Vimeo-90K. For BD degradation, we train it on Vimeo-90K and test it on Vimeo-90K-T, Vid4, and UDM10 [207]. The comparisons with existing methods are shown in Table 5.1. As we can see, RVRT achieves the best performance on REDS4 and Vid4 for both degradations. Compared with the representative recurrent model BasicVSR++ [177], RVRT improves the PSNR by significant margins of 0.2∼0.5dB. Compared with the recent transformer-based model VRT [2], RVRT outperforms VRT on REDS4 and Vid4 by up to 0.36dB. The visual comparisons of different methods are shown in Fig. 5.4. It is clear that RVRT generates sharp and clear HQ frames, while other methods fail to restore fine textures and details.

We compare the model size, testing memory consumption and runtime of different models in Table 5.2. Compared with representative

Table 5.1: Quantitative comparison (average PSNR/SSIM) with state-of-the-art methods for video super-resolution (×4) on REDS4 [237], Vimeo-90K-T [215], Vid4 [238] and UDM10 [207].

| Method | BI degradation | | | BD degradation | | |
|---|---|---|---|---|---|---|
| | REDS4 [237] (RGB channel) | Vimeo-90K-T [215] (Y channel) | Vid4 [238] (Y channel) | UDM10 [207] (Y channel) | Vimeo-90K-T [215] (Y channel) | Vid4 [238] (Y channel) |
| Bicubic | 26.14/0.7292 | 31.32/0.8684 | 23.78/0.6347 | 28.47/0.8253 | 31.30/0.8687 | 21.80/0.5246 |
| SwinIR [1] | 29.05/0.8269 | 35.67/0.9287 | 25.68/0.7491 | 35.42/0.9380 | 34.12/0.9167 | 25.25/0.7262 |
| SwinIR-ft [1] | 29.24/0.8319 | 35.89/0.9301 | 25.69/0.7488 | 36.76/0.9467 | 35.70/0.9293 | 25.62/0.7498 |
| TOFlow [215] | 27.98/0.7990 | 33.08/0.9054 | 25.89/0.7651 | 36.26/0.9438 | 34.62/0.9212 | 25.85/0.7659 |
| FRVSR [187] | - | - | - | 37.09/0.9522 | 35.64/0.9319 | 26.69/0.8103 |
| DUF [221] | 28.63/0.8251 | - | 27.33/0.8319 | 38.48/0.9605 | 36.87/0.9447 | 27.38/0.8329 |
| PFNL [207] | 29.63/0.8502 | 36.14/0.9363 | 26.73/0.8029 | 38.74/0.9627 | - | 27.16/0.8355 |
| RBPN [240] | 30.09/0.8590 | 37.07/0.9435 | 27.12/0.8180 | 38.66/0.9596 | 37.20/0.9458 | 27.17/0.8205 |
| MuCAN [171] | 30.88/0.8750 | 37.32/0.9465 | - | - | - | - |
| RLSP [173] | - | - | - | 38.48/0.9606 | 36.49/0.9403 | 27.48/0.8388 |
| TGA [182] | - | - | - | 38.74/0.9627 | 37.59/0.9516 | 27.63/0.8423 |
| RSDN [175] | - | - | - | 39.35/0.9653 | 37.23/0.9471 | 27.92/0.8505 |
| RRN [189] | - | - | - | 38.96/0.9644 | - | 27.69/0.8488 |
| FDAN [190] | - | - | - | 39.91/0.9686 | 37.75/0.9522 | 27.88/0.8508 |
| EDVR [24] | 31.09/0.8800 | 37.61/0.9489 | 27.35/0.8264 | 39.89/0.9686 | 37.81/0.9523 | 27.85/0.8503 |
| GOVSR [202] | - | - | - | 40.14/0.9713 | 37.63/0.9503 | 28.41/0.8724 |
| VSRT [8] | 31.19/0.8815 | 37.71/0.9494 | 27.36/0.8258 | - | - | - |
| BasicVSR [176] | 31.42/0.8909 | 37.18/0.9450 | 27.24/0.8251 | 39.96/0.9694 | 37.53/0.9498 | 27.96/0.8553 |
| IconVSR [176] | 31.67/0.8948 | 37.47/0.9476 | 27.39/0.8279 | 40.03/0.9694 | 37.84/0.9524 | 28.04/0.8570 |
| VRT [2] | 32.19/0.9006 | 38.20/0.9530 | 27.93/0.8425 | 41.05/0.9737 | 38.72/0.9584 | 29.42/0.8795 |
| BasicVSR++ [177] | 32.39/0.9069 | 37.79/0.9500 | 27.79/0.8400 | 40.72/0.9722 | 38.21/0.9550 | 29.04/0.8753 |
| **RVRT (ours)** | 32.75/0.9113 | 38.15/0.9527 | 27.99/0.8462 | 40.90/0.9729 | 38.59/0.9576 | 29.54/0.8810 |

Table 5.2: Comparison of model size, testing memory and runtime for an LQ input of 320 × 180.

| Method | #Param (M) | Memory (M) | Runtime (ms) | PSNR (dB) |
|---|---|---|---|---|
| BasicVSR++ [177] | 7.3 | 223 | 77 | 32.39 |
| BasicVSR++ [177]+RSTB [1] | 9.3 | 1021 | 201 | 32.61 |
| EDVR [24] | 20.6 | 3535 | 378 | 31.09 |
| VSRT [8] | 32.6 | 27487 | 328 | 31.19 |
| VRT [2] | 35.6 | 2149 | 243 | 32.19 |
| **RVRT (ours)** | 10.8 | 1056 | 183 | 32.75 |

parallel methods EDVR [24], VSRT [8] and VST [2], RVRT achieves significant performance gains with less than at least 50% of model parameters and testing memory usage. It also reduces the runtime by at least 25%. Compared the recurrent model BasicVSR++ [177], RVRT brings a PSNR improvement of 0.26dB. As for the inferiority of testing memory and runtime, we argue that it is mainly because the CNN layers are highly optimized on existing deep learning frameworks. To

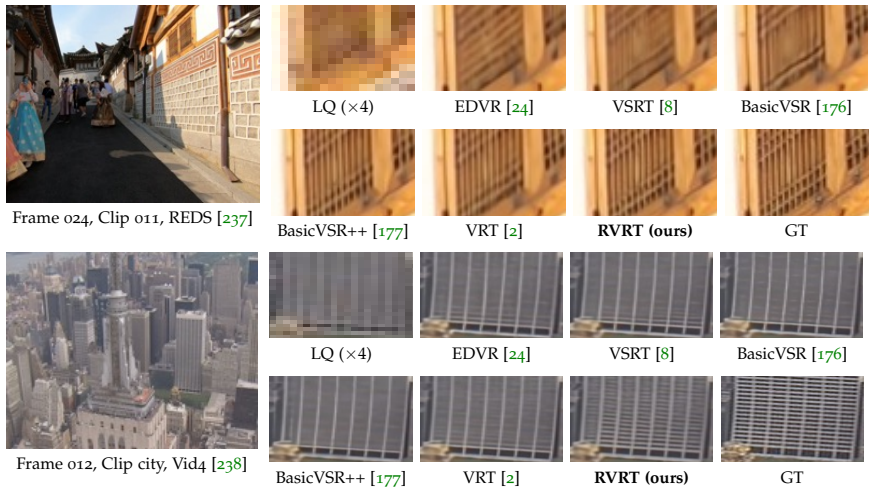



Figure 5.4: Visual comparison of video super-resolution (×4) methods on REDS [237] and Vid4 [238].

prove it, we use the transformer-based RSTB blocks in RVRT to replace the CNN blocks in BasicVSR++, in which case it has similar memory usage and more runtime than our model.

In addition, to better understand how guided deformable attention works, we visualize the predicted offsets on the LQ frames and show the attention weight in Fig. 5.5. As we can see, multiple offsets are predicted to select multiple sampled locations in the neighbourhood of the corresponding pixel. According to the feature similarity between the query feature and the sampled features, features of different locations are aggregated by calculating a dynamic attention weight.

### 5.4.3 *Video Deblurring*

For video deblurring, the model is trained and tested on two different datasets, DVD [180] and GoPro [21], with their official training/testing splits. As shown in Table 5.3 and 5.4, RVRT shows its superiority over most methods with huge improvements of 1.40∼2.27dB on two datasets. Even though the performance gain over VRT is relatively small, RVRT has a smaller model size and much less runtime. In detail, the model size and runtime of RVRT are 13.6M and 0.3s, while VRT has 18.3M parameters and the runtime of 2.2s on a $1280 \times 720$ LQ input. The

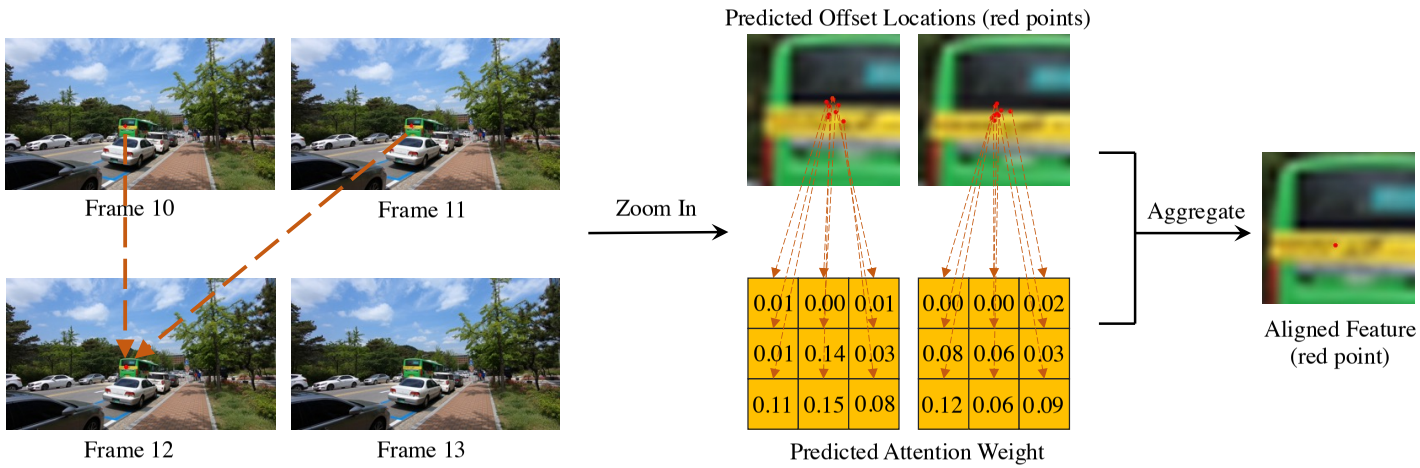



Figure 5.5: The visualization of predicted offsets and attention weights predicted in guided deformable attention. Although guided deformable attention is conducted on features, we plot illustrations on LQ input frames for better understanding. Best viewed by zooming.

Table 5.3: Quantitative comparison (average RGB channel PSNR/SSIM) with state-of-the-art methods for video deblurring on DVD [180].

| Method | DBN [180] | STFAN [181] | STTN [222] | SFE [208] | EDVR [24] | TSP [210] |
|---|---|---|---|---|---|---|
| PSNR | 30.01 | 31.24 | 31.61 | 31.71 | 31.82 | 32.13 |
| SSIM | 0.8877 | 0.9340 | 0.9160 | 0.9160 | 0.9160 | 0.9268 |
| Method | PVDNet [186] | GSTA [28] | ARVo [183] | FGST [262] | VRT [2] | **RVRT (ours)** |
| PSNR | 32.31 | 32.53 | 32.80 | 33.36 | 34.24 | 34.30 |
| SSIM | 0.9260 | 0.9468 | 0.9352 | 0.9500 | 0.9651 | 0.9655 |

Table 5.4: Quantitative comparison (average RGB channel PSNR/SSIM) with state-of-the-art methods for video deblurring on GoPro [21].

| Method | SRN [241] | MPRNet [244] | MAXIM [269] | IFI-RNN [184] | ESTRNN [185] | EDVR [24] |
|---|---|---|---|---|---|---|
| PSNR | 30.26 | 32.66 | 32.86 | 31.05 | 31.07 | 31.54 |
| SSIM | 0.9342 | 0.9590 | 0.9610 | 0.9110 | 0.9023 | 0.9260 |
| Method | TSP [210] | PVDNet [186] | GSTA [28] | FGST [262] | VRT [2] | **RVRT (ours)** |
| PSNR | 31.67 | 31.98 | 32.10 | 32.90 | 34.81 | 34.92 |
| SSIM | 0.9279 | 0.9280 | 0.9600 | 0.9610 | 0.9724 | 0.9738 |

visual comparison is provided in the supplementary material due to the space limit.

### 5.4.4 *Video Denoising*

For video denoising, we train the model on the training set of DAVIS [239] and test it on its corresponding testset and Set8 [25]. For fairness of comparison, following [25], [26], we train a non-blind additive white Gaussian denoising model for noise level $\sigma \sim \mathcal{U}(0, 50)$. Similar to the case of video deblurring, there is a huge gap (0.60∼2.37dB) between RVRT and most methods. Compared with VRT, RVRT has slightly better performance on large noise levels, with a smaller model size (12.8M *v.s.*18.4M) and less runtime (0.2s *v.s.*1.5s) on a $1280 \times 720$ LQ input. The visual comparison is provided in the supplementary material due to the space limit.

Table 5.5: Quantitative comparison (average RGB channel PSNR) with state-of-the-art methods for video denoising on DAVIS [239] and Set8 [25].

| Dataset | $\sigma$ | VLNB [245] | DVDNet [25] | FastDVDNet [26] | PaCNet [204] | VRT [2] | **RVRT (ours)** |
|---|---|---|---|---|---|---|---|
| DAVIS | 10 | 38.85 | 38.13 | 38.71 | 39.97 | 40.82 | 40.57 |
| | 20 | 35.68 | 35.70 | 35.77 | 36.82 | 38.15 | 38.05 |
| | 30 | 33.73 | 34.08 | 34.04 | 34.79 | 36.52 | 36.57 |
| | 40 | 32.32 | 32.86 | 32.82 | 33.34 | 35.32 | 35.47 |
| | 50 | 31.13 | 31.85 | 31.86 | 32.20 | 34.36 | 34.57 |
| Set8 | 10 | 37.26 | 36.08 | 36.44 | 37.06 | 37.88 | 37.53 |
| | 20 | 33.72 | 33.49 | 33.43 | 33.94 | 35.02 | 34.83 |
| | 30 | 31.74 | 31.79 | 31.68 | 32.05 | 33.35 | 33.30 |
| | 40 | 30.39 | 30.55 | 30.46 | 30.70 | 32.15 | 32.21 |
| | 50 | 29.24 | 29.56 | 29.53 | 29.66 | 31.22 | 31.33 |

### 5.4.5 *Ablation Study*

To explore the effectiveness of different components, we conduct ablation studies on REDS [237] for video SR. For efficiency, we reduce the MRSTB blocks by half and use 12 frames in training.

THE IMPACT OF CLIP LENGTH. In RVRT, we divide the video into $N$-frame clips. As shown in Table 5.6, the performance rises when clip length is increased from 1 to 2. However, the performance saturates when $N = 3$, possibly due to large within-clip motions and inaccurate optical flow derivation. When we directly estimate all optical flows (marked by $^*$), the PSNR hits 32.21dB. Besides, to compare the temporal modelling ability, we hack the input LQ video (Clip 000 from REDS, 100 frames in total) by manually setting all pixels of the 50-th frame as zeros. As indicated in Fig. 5.6, on the one hand, $N = 2$ has a smaller performance drop and all its frames still have higher PSNR than $N = 1$ (equals to a recurrent model) after the attack, showing that RVRT can mitigate the noise amplification from the hacked frame to the rest frames. One the other hand, the hacked frame of $N = 2$ has an impact on more neighbouring frames than $N = 1$, which means that RVRT can alleviate information loss and utilize more frames than $N = 1$ for restoration.

THE IMPACT OF VIDEO ALIGNMENT. The alignment of video clips plays a key role in our framework. We compare the proposed clip-to-clip guided deformable attention (GDA) with existing frame-to-frame

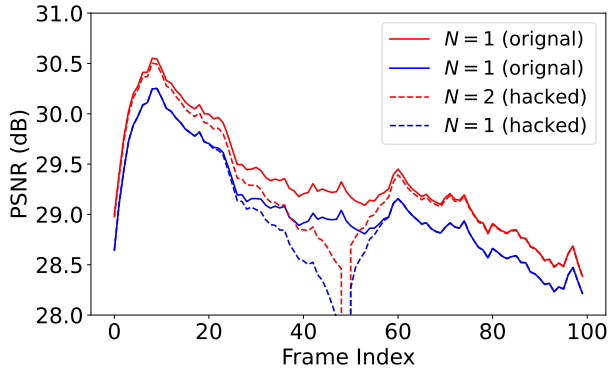



Figure 5.6: The robustness to noise injection attack with different clip lengths. It shows per-frame PSNR drop when pixels of the 50-th frame is hacked to be all zeros. $N$ is clip length.

Table 5.6: Ablation study on clip length.

| Clip | 1 | 2 | 3 | $3^*$ |
|---|---|---|---|---|
| PSNR | 31.98 | 32.10 | 32.07 | 32.21 |

Table 5.7: Ablation study on different video alignment techniques.

| Alignment | Warping [215] | TMSA [2] | DCN [172] | GDA* | GDA |
|---|---|---|---|---|---|
| PSNR | 28.88 | 30.45 | 31.93 | 32.00 | 32.10 |

alignment techniques by performing them frame by frame, followed by concatenation and channel reduction. As we can see from Table 5.7, GDA outperforms all existing methods when it is used for frame-to-frame alignment (denoted as GDA*), and leads a further improvement when we aggregate features directly from the whole clip.

THE IMPACT OF DIFFERENT COMPONENTS IN GDA. We further conduct an ablation study on GDA in Table 5.8. As we can see, the optical flow guidance is critical for the model, leading to a PSNR gain of 1.11dB. The update of optical flow in different layers can further improve the result. The channel interaction in MLP also plays an important role, since the attention mechanism only aggregates information spatially.

Table 5.8: Ablation study on different GDA components.

| | | | | |
|---|---|---|---|---|
| Optical Flow Guidance | | ✓ | ✓ | ✓ |
| Optical Flow Update | | | ✓ | ✓ |
| MLP | ✓ | ✓ | | ✓ |
| PSNR | 30.99 | 32.03 | 31.83 | 32.10 |

Table 5.9: Ablation study on deformable groups and attention heads.

| | | | | | | |
|---|---|---|---|---|---|---|
| Deformable Group | 1 | 6 | 12 | 12 | 12 | 24 |
| Attention Head | 1 | 6 | 12 | 24 | 36 | 24 |
| PSNR | 31.63 | 32.03 | 32.10 | 32.13 | 32.03 | 32.11 |

THE IMPACT OF DEFORMABLE GROUP AND ATTENTION HEAD. We also conduct experiments on different group and head numbers in GDA. As shown in Table 5.9, when the deformable group rises, the PSNR first rises and then keeps almost unchanged. Besides, double attention heads lead to slightly better results at the expense of higher computation, but using too many heads has an adverse impact as the head dimension may be too small.

## 5.5 CONCLUSION

In this chapter, we proposed a recurrent video restoration transformer with guided deformable attention. It is a globally recurrent model with locally parallel designs, which benefits from the advantages of both parallel methods and recurrent methods. We also propose the guided deformable attention module for our special case of video clip-to-clip alignment. Under the guidance of optical flow, it aggregates information from multiple neighboring locations adaptively with the attention mechanism. Extensive experiments on video super-resolution, video deblurring, and video denoising demonstrated the effectiveness of the proposed method.

# 6 CONCLUSION AND DISCUSSION

In this chapter, we will first summarize the contributions of this thesis and discuss the limitations of proposed methods. Then, social impacts and future work will be discussed.

## 6.1 SUMMARY

This thesis addressed the image and video restoration problems, including image/video super-resolution, deblurring, denoising, video frame interpolation and JPEG compression artifact reduction, *etc*. In total, we proposed a practical degradation model for real-world image super-resolution, a unified model for image restoration and two unified models for video restoration.

In Chapter 2, we proposed a practical degradation model that can simulate complicated real-world degradations. It is consisted of random shuffling of simple gradations, including Gaussian blur with different kernel sizes and deviations, downsampling with random scales and interpolation modes, and complex noise choices (Gaussian noise, JPEG compression noise and Processed camera sensor noise). Based on this degradation model, we trained a deep blind model for general image super-resolution and achieved good performance on both synthetic and real image datasets under diverse degradations. To the best of our knowledge, this is the first work to adopt a new hand-designed degradation model for general blind image super-resolution problem. It provides a new and practical way towards real-world image super-resolution applications.

In Chapter 3, considering that existing image restoration backbones still have limited performance as a result of convolution-based architectures, we proposed a new transformer-based framework SwinIR for different restoration tasks. It refines pixel features by the attention mechanism within a small partitioned window, achieving a good tradeoff between performance and efficiency. Since SwinIR allows for content-based interaction and long-range dependency modelling ability, the proposed method achieved significant improvements on image

super-resolution, denoising and compression artifact reduction, showing great potential and generalizability as a unified backbone model for different image restoration tasks.

Inspired by the good performance improvements in Chapter 3, we extended SwinIR to video restoration in Chapter 4 by taking an extra temporal dimension into consideration and name it as VRT. We extend the 2D attention to 3D attention and extract features at multiple scales. At each scale, we jointly extracts, aligns and fuses frame features by reciprocal attention and parallel warping, allowing for parallel frame prediction and long-range temporal dependency modelling abilities. Experimental results on benchmark datasets saw large PSNR improvements up to 2.16dB.

Although VRT achieved significant better performance in different video restoration tasks, it suffers from large model size and heavy computation. In Chapter 5, we continue to improve the video restoration model VRT by introducing a more lightweight model, namely, RVRT. It divides the video into small clips and uses a globally recurrent but locally parallel design for feature alignment and refinement. Information is accumulated and transmitted with a larger hidden state, which alleviates information loss and noise amplification in the recurrent architecture. Besides, the guided deformable attention is proposed accordingly for clip-to-clip alignment. Extensive results show that RVRT achieves state-of-the-art performance with balanced model size, memory usage and runtime.

## 6.2 LIMITATIONS

Although the proposed methods have boosted the performance in image and video restoration, they still have their own limitations. First, in the practical degradation model, while we have included several common degradation types, there are certain degradations that have not been explicitly considered. As mentioned in Chapter 1, other factors such as non-ideal lens and sensor-specific noises might also have significant impacts during the imaging process. Besides, a random shuffling of different degradations might not be enough to simulate the real cases. Second, in the image restoration model SwinIR, one problem is the computation efficiency. Although it improves the performance by large margins, it is about four times slower than the state-of-the-art CNN models and costs double testing memory. In particular, for image

denoising and compression artifact reduction, this framework refines image features at the same scale as the input image, which might be more computationally expensive than other multi-scale architectures. Third, for the video restoration model VRT, the biggest drawback is the heavy computation burden. As a parallel model, VRT deals with multiple frames at the same time, leading to extensive memory usage. Moreover, the used 3D attention consumes great amount of memory, although we have divided the video into small 3D windows. To remedy this problem, RVRT is proposed to reduce the memory usage by incorporating a recurrent framework. It significantly reduces the memory usage, but still has some other limitations. For example, the complexity of pre-alignment by optical flow increases quadratically with respect to the clip length.

## 6.3 SOCIETAL IMPACTS

While image and video restoration technologies offer significant benefits, it is crucial to be aware of the potential negative consequences and ethical challenges they pose. On the one hand, visual restoration may generate results with artifacts or even hallucinate the details in some cases. When the restored images or videos are used for later processing, it may lead to inaccurate results. For example, to alleviate the corruption of information, restoration techniques are used in forensic investigations, medical diagnosis or video surveillance. The reconstructed results might be misleading if the objects change their identities (especially for human faces) or the textures are synthesized in a low-fidelity way during restoration. In particular, from simple visual recognition methods to autonomous driving systems, most of them are trained on natural datasets that have rarely seen artifacts generated by restoration algorithms. The usage of restoration might significantly deteriorate their performance and cause serious decision errors. On the other hand, when the restored images or videos are of high fidelity, it brings other problems of information leaking and privacy concerns. For instance, the restoration of low-quality (*e.g.*, deliberately blurred images) may inadvertently reveal sensitive or private information that was originally obscured. In addition, restoration also suffers from other common problems of machine learning, such as model bias and discrimination.

## 6.4 FUTURE WORK

Considering the inherent challenges in visual restoration and the existing limitations of the proposed methods, there are several promising research directions that warrant further exploration in the future.

1. Real-world degradation model. Chapter 2 proposed a promising degradation model towards solving real-world image super-resolution, but it still cannot cover all degradation types. To be closer to the real scenarios, it is important to consider more elementary degradation operators and design a more complicated composition method, starting from the understanding of the imaging process.

2. Efficient image and video restoration. For real-world applications, especially for edge devices, the computation burden and inference latency are critical factors aside from reconstruction performance. At present, most methods suffer from high memory consumption and long latency, especially for the transformer-based methods. Although Chapter 5 has taken a step towards more efficient video restoration, the need for further research remains evident.

3. Better perceptual metrics for images and videos. A good perceptual metric provides a way for convenient and fair performance evaluation, and sometimes could be directly used as the optimization target. Finding better perceptual metrics that align well with humans is a fundamental problem, which might be more urgent than model design in the era of deep learning, although it is often regarded as a separate research field of image/video quality assessment.

4. Restoration with auxiliary input. Merely using the low-quality image/video input might have its upper bound of performance. Finding an auxiliary input can provide more information and might facilitate the restoration process. There are some work on this direction, such as reference-based image super-resolution [9], event-assisted deblurring [11] and video frame interpolation [14]. How to encode auxiliary input and fuse it with existing frameworks effectively still needs more research.

5. Interpretability. The deep neural networks are like black boxes at the moment. We do not understand how it works, making it difficult to improve the design and increase the robustness to extreme cases or possible attacks. Aside from [275]–[277], more attempts for understanding the neural network could be beneficial. Besides, we can try to design explainable restoration methods such as incorporating neural networks with traditional optimization-based methods [37], [61], [278].

6. Exploration of large model and large data. Recently, in the field of image generation, diffusion models have shown great breakthroughs in generating photo-realistic images [279], [280]. This could inspire our research in visual restoration in two aspects. The first is how to utilize the learned general image prior for restoration, and the second is how to train a powerful restoration model based on large model size and large data.

# BIBLIOGRAPHY


[1] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, „SwinIR: Image restoration using swin transformer," in *IEEE Conference on International Conference on Computer Vision Workshops*, 2021, pp. 1833–1844 (cit. on pp. ix, 1, 56, 57, 63, 69, 82, 85, 87, 94).

[2] J. Liang, J. Cao, Y. Fan, K. Zhang, R. Ranjan, Y. Li, R. Timofte, and L. Van Gool, „Vrt: A video restoration transformer," *arXiv preprint arXiv:2201.12288*, 2022 (cit. on pp. ix, 1, 81–85, 93–95, 97–99).

[3] J. Liang, Y. Fan, X. Xiang, R. Ranjan, E. Ilg, S. Green, J. Cao, K. Zhang, R. Timofte, and L. Van Gool, „Recurrent video restoration transformer with guided deformable attention," in *Advances in Neural Information Processing Systems*, 2022, pp. 378–393 (cit. on p. ix).

[4] K. Zhang, J. Liang, L. Van Gool, and R. Timofte, „Designing a practical degradation model for deep blind image super-resolution," in *IEEE Conference on International Conference on Computer Vision*, 2021, pp. 4791–4800 (cit. on pp. ix, 29, 37, 39, 43, 45, 56, 84).

[5] J. Liang, K. Zhang, S. Gu, L. Van Gool, and R. Timofte, „Flow-based kernel prior with application to blind super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 601–10 610 (cit. on pp. ix, 12, 56, 84).

[6] J. Liang, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, „Mutual affine network for spatially variant kernel estimation in blind image super-resolution," in *IEEE Conference on International Conference on Computer Vision*, 2021, pp. 4096–4105 (cit. on pp. ix, 4, 56, 84).

[7] J. Liang, A. Lugmayr, K. Zhang, M. Danelljan, L. Van Gool, and R. Timofte, „Hierarchical conditional flow: A unified framework for image super-resolution and image rescaling," in *IEEE Conference*

*on International Conference on Computer Vision*, 2021, pp. 4076–4085 (cit. on pp. ix, 9, 56, 84).

[8] J. Cao, Y. Li, K. Zhang, and L. Van Gool, „Video super-resolution transformer," *arXiv preprint arXiv:2106.06847*, 2021 (cit. on pp. x, 30, 33, 34, 55, 57, 68–70, 81, 82, 84, 85, 94, 95).

[9] J. Cao, J. Liang, K. Zhang, Y. Li, Y. Zhang, W. Wang, and L. V. Gool, „Reference-based image super-resolution with deformable attention transformer," in *European Conference on Computer Vision*, 2022, pp. 325–342 (cit. on pp. x, 104).

[10] J. Cao, J. Liang, K. Zhang, W. Wang, Q. Wang, Y. Zhang, H. Tang, and L. Van Gool, „Towards interpretable video super-resolution via alternating optimization," in *European Conference on Computer Vision*, 2022, pp. 393–411 (cit. on pp. x, 84).

[11] L. Sun, C. Sakaridis, J. Liang, Q. Jiang, K. Yang, P. Sun, Y. Ye, K. Wang, and L. V. Gool, „Event-based fusion for motion deblurring with cross-modal attention," in *European Conference on Computer Vision*, 2022, pp. 412–428 (cit. on pp. x, 104).

[12] K. Zhang, Y. Li, J. Liang, J. Cao, Y. Zhang, H. Tang, R. Timofte, and L. Van Gool, „Practical blind denoising via swin-conv-unet and data synthesis," *Machine Intelligence Research*, 2022 (cit. on pp. x, 84).

[13] J. Cao, Q. Wang, J. Liang, Y. Zhang, K. Zhang, and L. Van Gool, „Practical real video denoising with realistic degradation model," *arXiv preprint arXiv:2208.11803*, 2022 (cit. on pp. x, 85).

[14] L. Sun, C. Sakaridis, J. Liang, P. Sun, J. Cao, K. Zhang, Q. Jiang, K. Wang, and L. Van Gool, „Event-based frame interpolation with ad-hoc deblurring," in *Computer Vision and Pattern Recognition*, 2022, pp. 1146–1155 (cit. on pp. x, 104).

[15] Y. Li, K. Zhang, J. Liang, J. Cao, C. Liu, R. Gong, Y. Zhang, H. Tang, Y. Liu, D. Demandolx, *et al.*, „LSDIR: A large scale dataset for image restoration," in *Computer Vision and Pattern Recognition Workshops*, 2023, pp. 72–81 (cit. on p. x).

[16] Y. Zhu, K. Zhang, J. Liang, J. Cao, B. Wen, R. Timofte, and L. Van Gool, „Denoising diffusion models for plug-and-play image restoration," *arXiv preprint arXiv:2305.08995*, 2023 (cit. on p. x).

[17] C. Dong, C. C. Loy, K. He, and X. Tang, „Learning a deep convolutional network for image super-resolution," in *European Conference on Computer Vision*, 2014, pp. 184–199 (cit. on pp. 1, 9, 10, 29, 32, 33, 56, 84).

[18] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, „Image super-resolution using very deep residual channel attention networks," in *European Conference on Computer Vision*, 2018, pp. 286–301 (cit. on pp. 1, 9, 12, 29, 32, 39–41, 43, 56, 77, 84).

[19] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, „Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017 (cit. on pp. 1, 19, 29, 33, 45–48, 56, 84).

[20] K. Zhang, W. Zuo, and L. Zhang, „Ffdnet: Toward a fast and flexible solution for cnn-based image denoising," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4608–4622, 2018 (cit. on pp. 1, 22, 29, 33, 45–47).

[21] S. Nah, T. Hyun Kim, and K. Mu Lee, „Deep multi-scale convolutional neural network for dynamic scene deblurring," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3883–3891 (cit. on pp. 1, 66, 67, 72–74, 92, 93, 95, 97).

[22] Y. Gandelsman, A. Shocher, and M. Irani, „Double-dip": Unsupervised image decomposition via coupled deep-image-priors," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 026–11 035 (cit. on p. 1).

[23] C. Dong, Y. Deng, C. C. Loy, and X. Tang, „Compression artifacts reduction by a deep convolutional network," in *IEEE International Conference on Computer Vision*, 2015, pp. 576–584 (cit. on pp. 1, 33, 48).

[24] X. Wang, K. C. Chan, K. Yu, C. Dong, and C. Change Loy, „Edvr: Video restoration with enhanced deformable convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 1954–1963 (cit. on pp. 1, 54, 56, 57, 66–70, 72–74, 77, 81–84, 93–95, 97).

[25] M. Tassano, J. Delon, and T. Veit, „Dvdnet: A fast network for deep video denoising," in *IEEE International Conference on Image Processing*, 2019, pp. 1805–1809 (cit. on pp. 1, 56, 66, 67, 74–76, 84, 93, 97, 98).

[26] M. Tassano, J. Delon, and T. Veit, „Fastdvdnet: Towards real-time deep video denoising without flow estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1354–1363 (cit. on pp. 1, 56, 66, 67, 74–76, 84, 97, 98).

[27] J. Pan, H. Bai, and J. Tang, „Cascaded deep video deblurring using temporal sharpness prior," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3043–3051 (cit. on pp. 1, 56).

[28] M. Suin and A. Rajagopalan, „Gated spatio-temporal attention-guided video deblurring," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7802–7811 (cit. on pp. 1, 57, 67, 72, 73, 93, 97).

[29] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller, and J. Kautz, „Super slomo: High quality estimation of multiple intermediate frames for video interpolation," in *IEEE conference on computer vision and pattern recognition*, 2018, pp. 9000–9008 (cit. on pp. 1, 76, 77).

[30] S. Niklaus, L. Mai, and F. Liu, „Video frame interpolation via adaptive separable convolution," in *IEEE International Conference on Computer Vision*, 2017, pp. 261–270 (cit. on pp. 1, 76, 77).

[31] L. Van Gool, G. Szekely, and V. Ferrari, *Computer Vision*. 2014, pp. 62–73 (cit. on p. 2).

[32] J. Telleen, A. Sullivan, J. Yee, O. Wang, P. Gunawardane, I. Collins, and J. Davis, „Synthetic shutter speed imaging," in *Computer Graphics Forum*, vol. 26, 2007, pp. 591–598 (cit. on p. 3).

[33] W. Shen, W. Bao, G. Zhai, L. Chen, X. Min, and Z. Gao, „Blurry video frame interpolation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5114–5123 (cit. on p. 3).

[34] T. F. Chan and C.-K. Wong, „Total variation blind deconvolution," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 370–375, 1998 (cit. on p. 5).

[35] K. He, J. Sun, and X. Tang, „Single image haze removal using dark channel prior," *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2341–2353, 2010 (cit. on pp. 5, 32).

[36] M. V. Afonso, J. M. Bioucas-Dias, and M. A. Figueiredo, „Fast image recovery using variable splitting and constrained optimization," *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2345–2356, 2010 (cit. on p. 5).

[37] K. Zhang, L. V. Gool, and R. Timofte, „Deep unfolding network for image super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3217–3226 (cit. on pp. 5, 10, 14, 16, 105).

[38] R. Timofte, V. De Smet, and L. Van Gool, „A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Asian Conference on Computer Vision*, 2014, pp. 111–126 (cit. on pp. 9, 30, 32).

[39] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, „Enhanced deep residual networks for single image super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 136–144 (cit. on pp. 9, 10, 21, 29, 32, 39, 42).

[40] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, „Esrgan: Enhanced super-resolution generative adversarial networks," in *European Conference on Computer Vision Workshops*, 2018, pp. 701–710 (cit. on pp. 9, 10, 21, 23, 26, 29, 32, 37, 39–41, 43, 45).

[41] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, „Deep laplacian pyramid networks for fast and accurate super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 624–632 (cit. on pp. 9, 12, 32).

[42] Z. Hui, X. Gao, Y. Yang, and X. Wang, „Lightweight image super-resolution with information multi-distillation network," in *ACM International Conference on Multimedia*, 2019, pp. 2024–2032 (cit. on pp. 9, 43, 44).

[43] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, and L. Zhang, „Ntire 2017 challenge on single image super-resolution: Methods and results," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 114–125 (cit. on pp. 10, 21).

[44] C. Liu and D. Sun, „On bayesian adaptive video super resolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 2, pp. 346–360, 2013 (cit. on pp. 10, 14–16).

[45] A. Shocher, N. Cohen, and M. Irani, „"zero-shot" super-resolution using deep internal learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3118–3126 (cit. on p. 10).

[46] N. Efrat, D. Glasner, A. Apartsin, B. Nadler, and A. Levin, „Accurate blur models vs. image priors in single image super-resolution," in *IEEE Conference on International Conference on Computer Vision*, 2013, pp. 2832–2839 (cit. on pp. 10, 14).

[47] S. Bell-Kligler, A. Shocher, and M. Irani, „Blind super-resolution kernel estimation using an internal-gan," in *Advances in Neural Information Processing Systems*, 2019, pp. 284–293 (cit. on pp. 10, 13, 16, 22, 23).

[48] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, „Residual dense network for image super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2472–2481 (cit. on pp. 10, 29, 32, 42, 56).

[49] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, *et al.*, „Photo-realistic single image super-resolution using a generative adversarial network," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4681–4690 (cit. on pp. 10, 21, 29, 32, 56).

[50] K. Zhang, W. Zuo, and L. Zhang, „Learning a single convolutional super-resolution network for multiple degradations," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3262–3271 (cit. on pp. 10, 12–14, 16, 29, 56, 84).

[51] J. Gu, H. Lu, W. Zuo, and C. Dong, „Blind super-resolution with iterative kernel correction," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1604–1613 (cit. on pp. 10, 13, 18, 19, 22–24).

[52] Y. Huang, S. Li, L. Wang, T. Tan, *et al.*, „Unfolding the alternating optimization for blind super resolution," *Advances in Neural Information Processing Systems*, pp. 5632–5643, 2020 (cit. on pp. 10, 13).

[53] X. Ji, Y. Cao, Y. Tai, C. Wang, J. Li, and F. Huang, „Real-world super-resolution via kernel estimation and noise injection," in *IEEE Conference on Computer Vision and Pattern Recognition Work-*

*shops*, 2020, pp. 466–467 (cit. on pp. 10, 13, 14, 23, 24, 26, 43, 45).

[54] M. S. Sajjadi, B. Schölkopf, and M. Hirsch, „Enhancenet: Single image super-resolution through automated texture synthesis," in *IEEE International Conference on Computer Vision*, 2017, pp. 4501–4510 (cit. on p. 12).

[55] T. Michaeli and M. Irani, „Nonparametric blind super-resolution," in *IEEE Conference on International Conference on Computer Vision*, 2013, pp. 945–952 (cit. on pp. 12, 32).

[56] K. Zhang, X. Zhou, H. Zhang, and W. Zuo, „Revisiting single image super-resolution under internet environment: Blur kernels and reconstruction algorithms," in *Pacific Rim Conference on Multimedia*, 2015, pp. 677–687 (cit. on p. 12).

[57] H. Zhang, Y. Dai, H. Li, and P. Koniusz, „Deep stacked hierarchical multi-patch network for image deblurring," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5978–5986 (cit. on pp. 12, 18, 19).

[58] K. Zhang, W. Zuo, S. Gu, and L. Zhang, „Learning deep cnn denoiser prior for image restoration," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3929–3938 (cit. on pp. 12, 14, 29, 45–47).

[59] W. Dong, L. Zhang, G. Shi, and X. Li, „Nonlocally centralized sparse representation for image restoration," *IEEE Transaction on Image Processing*, vol. 22, no. 4, pp. 1620–1630, 2013 (cit. on pp. 12, 14).

[60] T. Peleg and M. Elad, „A statistical prediction model based on sparse representations for single image super-resolution," *IEEE Transaction on Image Processing*, vol. 23, no. 6, pp. 2569–2582, 2014 (cit. on p. 12).

[61] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte, „Plug-and-play image restoration with deep denoiser prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021 (cit. on pp. 12, 29, 32, 33, 40, 45–48, 56, 105).

[62] T. Plotz and S. Roth, „Benchmarking denoising algorithms with real photographs," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1586–1595 (cit. on pp. 12, 18).

[63] Y. Yuan, S. Liu, J. Zhang, Y. Zhang, C. Dong, and L. Lin, „Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 701–710 (cit. on pp. 12–14).

[64] A. Lugmayr, M. Danelljan, and R. Timofte, „Unsupervised learning for real-world super-resolution," in *IEEE Conference on International Conference on Computer Vision Workshops*, 2019, pp. 3408–3416 (cit. on pp. 12–14).

[65] A. Shocher, N. Cohen, and M. Irani, „Zero-shot super-resolution using deep internal learning," in *IEEE International Conference on Computer Vision*, 2018, pp. 3118–3126 (cit. on pp. 13–15).

[66] V. Cornillere, A. Djelouah, W. Yifan, O. Sorkine-Hornung, and C. Schroers, „Blind image super-resolution with spatially variant degradations," *ACM TOG*, vol. 38, no. 6, pp. 1–13, 2019 (cit. on p. 13).

[67] J. Cai, H. Zeng, H. Yong, Z. Cao, and L. Zhang, „Toward real-world single image super-resolution: A new benchmark and a new model," in *IEEE Conference on International Conference on Computer Vision*, 2019, pp. 3086–3095 (cit. on p. 13).

[68] P. Wei, H. Lu, R. Timofte, L. Lin, W. Zuo, *et al.*, „Aim 2020 challenge on real image super-resolution: Methods and results," in *European Conference on Computer Vision Workshops*, 2020 (cit. on p. 13).

[69] L. Wang, Y. Wang, X. Dong, Q. Xu, J. Yang, W. An, and Y. Guo, „Unsupervised degradation representation learning for blind super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 581–10 590 (cit. on pp. 13, 56).

[70] A. Lugmayr, M. Danelljan, R. Timofte, N. Ahn, D. Bai, J. Cai, Y. Cao, J. Chen, K. Cheng, S. Chun, *et al.*, „Ntire 2020 challenge on real-world image super-resolution: Methods and results," in *IEEE Conference on International Conference on Computer Vision Workshops*, 2020, pp. 3408–3416 (cit. on pp. 14, 25).

[71] G. Riegler, S. Schulter, M. Ruther, and H. Bischof, „Conditioned regression models for non-blind single image super-resolution," in *IEEE Conference on International Conference on Computer Vision*, 2015, pp. 522–530 (cit. on pp. 14, 16).

[72] S. Park, E. Serpedin, and K. Qaraqe, „Gaussian assumption: The least favorable but the most useful [lecture notes],“ *IEEE SPM*, vol. 30, no. 3, pp. 183–186, 2013 (cit. on p. 17).

[73] S. Nam, Y. Hwang, Y. Matsushita, and S. J. Kim, „A holistic approach to cross-channel image noise modeling and its application to image denoising,“ in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1683–1691 (cit. on p. 17).

[74] T. Brooks, B. Mildenhall, T. Xue, J. Chen, D. Sharlet, and J. T. Barron, „Unprocessing images for learned raw denoising,“ in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 036–11 045 (cit. on p. 18).

[75] Adobe, „Digital negative specification,“ Version 1.5.00, 2019 (cit. on p. 18).

[76] H. S. Malvar, L.-w. He, and R. Cutler, „High-quality linear interpolation for demosaicing of bayer-patterned color images,“ in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, 2004, pp. iii–485 (cit. on p. 18).

[77] M. D. Grossberg and S. K. Nayar, „What is the space of camera response functions?“ In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003, pp. II–602 (cit. on p. 18).

[78] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, „Randaugment: Practical automated data augmentation with a reduced search space,“ in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 702–703 (cit. on p. 19).

[79] J. Jiang, K. Zhang, and R. Timofte, „Towards flexible blind JPEG artifacts removal,“ in *IEEE International Conference on Computer Vision*, 2021 (cit. on p. 21).

[80] Y. Blau and T. Michaeli, „The perception-distortion tradeoff,“ in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6228–6237 (cit. on p. 21).

[81] E. Agustsson and R. Timofte, „Ntire 2017 challenge on single image super-resolution: Dataset and study,“ in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 126–135 (cit. on pp. 21, 39, 48).

[82] K. Ma, Z. Duanmu, Q. Wu, Z. Wang, H. Yong, H. Li, and L. Zhang, „Waterloo exploration database: New challenges for image quality assessment models," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 1004–1016, 2016 (cit. on pp. 21, 40).

[83] T. Karras, S. Laine, and T. Aila, „A style-based generator architecture for generative adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4401–4410 (cit. on p. 21).

[84] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, „Image-to-image translation with conditional adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1125–1134 (cit. on p. 21).

[85] D. Martin, C. Fowlkes, D. Tal, and J. Malik, „A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *IEEE International Conference on Computer Vision*, 2001, pp. 416–423 (cit. on p. 22).

[86] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa, „Sketch-based manga retrieval using manga109 dataset," *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 21 811–21 838, 2017 (cit. on p. 22).

[87] A. Ignatov, N. Kobyshev, R. Timofte, K. Vanhoey, and L. Van Gool, „Dslr-quality photos on mobile devices with deep convolutional networks," in *IEEE Conference on International Conference on Computer Vision*, 2017, pp. 3277–3285 (cit. on p. 22).

[88] M. Fritsche, S. Gu, and R. Timofte, „Frequency separation for real-world super-resolution," in *IEEE Conference on International Conference on Computer Vision Workshops*, 2019, pp. 3599–3608 (cit. on pp. 23, 24, 26, 29).

[89] A. Mittal, R. Soundararajan, and A. C. Bovik, „Making a "completely blind" image quality analyzer," *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 209–212, 2012 (cit. on pp. 25, 27).

[90] C. Ma, C.-Y. Yang, X. Yang, and M.-H. Yang, „Learning a no-reference quality metric for single-image super-resolution," *Computer Vision and Image Understanding*, vol. 158, pp. 1–16, 2017 (cit. on pp. 25, 27).

[91] Y. Blau, R. Mechrez, R. Timofte, T. Michaeli, and L. Zelnik-Manor, „The 2018 PIRM challenge on perceptual image super-resolution," in *European Conference on Computer Vision Workshops*, 2018 (cit. on pp. 25, 27).

[92] G. Jinjin, C. Haoming, C. Haoyu, Y. Xiaoxing, J. S. Ren, and D. Chao, „Pipal: A large-scale image quality assessment dataset for perceptual image restoration," in *European Conference on Computer Vision*, 2020, pp. 633–651 (cit. on p. 25).

[93] J. Kim, J. Kwon Lee, and K. Mu Lee, „Accurate image super-resolution using very deep convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1646–1654 (cit. on pp. 29, 32, 42).

[94] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, and W. Wu, „Feedback network for image super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3867–3876 (cit. on p. 29).

[95] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, „Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on image processing*, vol. 16, no. 8, pp. 2080–2095, 2007 (cit. on pp. 30, 45–47).

[96] S. Gu, N. Sang, and F. Ma, „Fast image super resolution via local regression," in *IEEE Conference on International Conference on Pattern Recognition*, 2012, pp. 3128–3131 (cit. on pp. 30, 32).

[97] K. He, X. Zhang, S. Ren, and J. Sun, „Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778 (cit. on pp. 30, 32).

[98] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, „Rethinking the inception architecture for computer vision," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826 (cit. on p. 30).

[99] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, „Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017 (cit. on pp. 30, 32, 33, 37, 38, 55, 57, 60, 82, 85, 91).

[100] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, „End-to-end object detection with transformers," in *European Conference on Computer Vision*, 2020, pp. 213–229 (cit. on pp. 30, 33, 57, 85).

[101] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, „Training data-efficient image transformers & distillation through attention," *arXiv preprint arXiv:2012.12877*, 2020 (cit. on pp. 30, 33).

[102] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, „An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020 (cit. on pp. 30, 33, 57, 85).

[103] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, „Swin transformer: Hierarchical vision transformer using shifted windows," *arXiv preprint arXiv:2103.14030*, 2021 (cit. on pp. 30, 33, 34, 37, 38, 57, 62, 63, 82, 85).

[104] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, „Pre-trained image processing transformer," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 299–12 310 (cit. on pp. 30, 33, 40–42, 47, 57, 85).

[105] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, „Stand-alone self-attention in vision models," *arXiv preprint arXiv:1906.05909*, 2019 (cit. on pp. 30, 33).

[106] A. Vaswani, P. Ramachandran, A. Srinivas, N. Parmar, B. Hechtman, and J. Shlens, „Scaling local self-attention for parameter efficient visual backbones," *arXiv preprint arXiv:2103.12731*, 2021 (cit. on pp. 30–33, 37, 85).

[107] J.-B. Cordonnier, A. Loukas, and M. Jaggi, „On the relationship between self-attention and convolutional layers," *arXiv preprint arXiv:1911.03584*, 2019 (cit. on p. 31).

[108] G. Elsayed, P. Ramachandran, J. Shlens, and S. Kornblith, „Revisiting spatial invariance with low-rank local connectivity," in *International Conference on Machine Learning*, 2020, pp. 2868–2879 (cit. on pp. 31, 37).

[109] R. Timofte, V. De Smet, and L. Van Gool, „Anchored neighborhood regression for fast example-based super-resolution," in *IEEE Conference on International Conference on Computer Vision*, 2013, pp. 1920–1927 (cit. on p. 32).

[110] L. Cavigelli, P. Hager, and L. Benini, „Cas-cnn: A deep convolutional neural network for image compression artifact suppression," in *2017 International Joint Conference on Neural Networks*, 2017, pp. 752–759 (cit. on pp. 32, 33).

[111] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, „Residual dense network for image restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 7, pp. 2480–2495, 2020 (cit. on pp. 32, 33, 40, 47, 48).

[112] K. Simonyan and A. Zisserman, „Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014 (cit. on p. 32).

[113] Y. Tai, J. Yang, X. Liu, and C. Xu, „Memnet: A persistent memory network for image restoration," in *IEEE International Conference on Computer Vision*, 2017, pp. 4539–4547 (cit. on pp. 32, 33).

[114] M. Haris, G. Shakhnarovich, and N. Ukita, „Deep back-projection networks for super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1664–1673 (cit. on pp. 32, 39–41).

[115] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, „Second-order attention network for single image super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 065–11 074 (cit. on pp. 32, 42).

[116] S. Zhou, J. Zhang, W. Zuo, and C. C. Loy, „Cross-scale internal graph neural network for image super-resolution," *arXiv preprint arXiv:2006.16673*, 2020 (cit. on pp. 32, 40–42).

[117] B. Niu, W. Wen, W. Ren, X. Zhang, L. Yang, S. Wang, K. Zhang, X. Cao, and H. Shen, „Single image super-resolution via a holistic attention network," in *European Conference on Computer Vision*, 2020, pp. 191–207 (cit. on pp. 32, 39–41).

[118] Y. Mei, Y. Fan, and Y. Zhou, „Image super-resolution with non-local sparse attention," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3517–3526 (cit. on pp. 32, 39–41, 56).

[119] D. Liu, B. Wen, Y. Fan, C. C. Loy, and T. S. Huang, „Non-local recurrent network for image restoration," *arXiv preprint arXiv:1806.02919*, 2018 (cit. on pp. 32, 33, 45, 46, 56).

[120] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, „Residual non-local attention networks for image restoration," *arXiv preprint arXiv:1903.10082*, 2019 (cit. on pp. 32, 33, 42, 45–48, 56).

[121] Y. Mei, Y. Fan, Y. Zhou, L. Huang, T. S. Huang, and H. Shi, „Image super-resolution with cross-scale non-local attention and exhaustive self-exemplars mining," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5690–5699 (cit. on pp. 32, 56).

[122] X. Mao, C. Shen, and Y.-B. Yang, „Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," *Advances in neural information processing systems*, vol. 29, pp. 2802–2810, 2016 (cit. on p. 33).

[123] T. Plötz and S. Roth, „Neural nearest neighbors networks," *arXiv preprint arXiv:1810.12575*, 2018 (cit. on pp. 33, 45, 46).

[124] Y. Peng, L. Zhang, S. Liu, X. Wu, Y. Zhang, and X. Wang, „Dilated residual networks with symmetric skip connection for image denoising," *Neurocomputing*, vol. 345, pp. 67–76, 2019 (cit. on pp. 33, 47).

[125] X. Jia, S. Liu, X. Feng, and L. Zhang, „Focnet: A fractional optimal control network for image denoising," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6054–6063 (cit. on pp. 33, 45, 46).

[126] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, „Multi-level wavelet-cnn for image restoration," in *IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 773–782 (cit. on pp. 33, 45, 46).

[127] X. Liu, X. Wu, J. Zhou, and D. Zhao, „Data-driven sparsity-based restoration of jpeg-compressed images in dual transform-pixel domain," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5171–5178 (cit. on p. 33).

[128] X. Zhang, W. Yang, Y. Hu, and J. Liu, „Dmcnn: Dual-domain multi-scale convolutional neural network for compression artifacts removal," in *IEEE International Conference on Image Processing*, 2018, pp. 390–394 (cit. on p. 33).

[129] B. Zheng, Y. Chen, X. Tian, F. Zhou, and X. Liu, „Implicit dual-domain convolutional network for robust color image compression artifact reduction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 11, pp. 3982–3994, 2019 (cit. on p. 33).

[130] M. Ehrlich, L. Davis, S.-N. Lim, and A. Shrivastava, „Quantization guided jpeg artifact correction," in *European Conference on Computer Vision*, 2020, pp. 293–309 (cit. on pp. 33, 48).

[131] B. Wu, C. Xu, X. Dai, A. Wan, P. Zhang, Z. Yan, M. Tomizuka, J. Gonzalez, K. Keutzer, and P. Vajda, „Visual transformers: Token-based image representation and processing for computer vision," *arXiv preprint arXiv:2006.03677*, 2020 (cit. on pp. 33, 85).

[132] Y. Li, K. Zhang, J. Cao, R. Timofte, and L. Van Gool, „Localvit: Bringing locality to vision transformers," *arXiv preprint arXiv:2104.05707*, 2021 (cit. on pp. 33, 57, 85).

[133] Y. Liu, G. Sun, Y. Qiu, L. Zhang, A. Chhatkuli, and L. Van Gool, „Transformer in convolutional neural networks," *arXiv preprint arXiv:2106.03180*, 2021 (cit. on pp. 33, 57).

[134] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, „Deep learning for generic object detection: A survey," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 261–318, 2020 (cit. on pp. 33, 57, 85).

[135] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr, *et al.*, „Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6881–6890 (cit. on p. 33).

[136] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, „Swin-unet: Unet-like pure transformer for medical image segmentation," *arXiv preprint arXiv:2105.05537*, 2021 (cit. on p. 33).

[137] D. Liang, X. Chen, W. Xu, Y. Zhou, and X. Bai, „Transcrowd: Weakly-supervised crowd counting with transformer," *arXiv preprint arXiv:2104.09116*, 2021 (cit. on p. 33).

[138] G. Sun, Y. Liu, T. Probst, D. P. Paudel, N. Popovic, and L. Van Gool, „Boosting crowd counting with transformers," *arXiv preprint arXiv:2105.10926*, 2021 (cit. on pp. 33, 57, 85).

[139] Z. Wang, X. Cun, J. Bao, and J. Liu, „Uformer: A general u-shaped transformer for image restoration," *arXiv preprint arXiv:2106.03106*, 2021 (cit. on pp. 33, 34, 57, 85).

[140] T. Xiao, M. Singh, E. Mintun, T. Darrell, P. Dollár, and R. Girshick, „Early convolutions help transformers see better," *arXiv preprint arXiv:2106.14881*, 2021 (cit. on p. 34).

[141] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, „Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1874–1883 (cit. on pp. 36, 59, 87).

[142] X. Wang, L. Xie, C. Dong, and Y. Shan, „Real-esrgan: Training real-world blind super-resolution with pure synthetic data," *arXiv preprint arXiv:2107.10833*, 2021 (cit. on pp. 37, 40, 43, 45).

[143] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, „Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680 (cit. on p. 37).

[144] J. Johnson, A. Alahi, and L. Fei-Fei, „Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*, Springer, 2016, pp. 694–711 (cit. on p. 37).

[145] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, „Two deterministic half-quadratic regularization algorithms for computed imaging," in *International Conference on Image Processing*, 1994, pp. 168–172 (cit. on pp. 37, 59, 87).

[146] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, and L. Zhang, „Ntire 2017 challenge on single image super-resolution: Methods and results," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 114–125 (cit. on p. 39).

[147] I. Loshchilov and F. Hutter, „Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017 (cit. on p. 39).

[148] D. P. Kingma and J. Ba, „Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014 (cit. on pp. 39, 65, 92).

[149] X. Wang, K. Yu, C. Dong, and C. C. Loy, „Recovering realistic texture in image super-resolution by deep spatial feature transform," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 606–615 (cit. on p. 39).

[150] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, „Contour detection and hierarchical image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 5, pp. 898–916, 2010 (cit. on p. 40).

[151] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, „Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004 (cit. on p. 40).

[152] C. Yim and A. C. Bovik, „Quality assessment of deblocked images," *IEEE Transactions on Image Processing*, vol. 20, no. 1, pp. 88–98, 2010 (cit. on p. 40).

[153] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-l. A. Morel, „Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *British Machine Vision Conference*, 2012, pp. 135.1–135.10 (cit. on pp. 41, 44).

[154] R. Zeyde, M. Elad, and M. Protter, „On single image scale-up using sparse-representations," in *International Conference on Curves and Surfaces*, 2010, pp. 711–730 (cit. on pp. 41, 44).

[155] D. Martin, C. Fowlkes, D. Tal, and J. Malik, „A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *IEEE Conference on International Conference on Computer Vision*, 2001, pp. 416–423 (cit. on pp. 41, 44).

[156] J.-B. Huang, A. Singh, and N. Ahuja, „Single image super-resolution from transformed self-exemplars," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5197–5206 (cit. on pp. 41, 44, 46, 47).

[157] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa, „Sketch-based manga retrieval using manga109 dataset," *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 21 811–21 838, 2017 (cit. on pp. 41, 44, 48, 50).

[158] X. He, Z. Mo, P. Wang, Y. Liu, M. Yang, and J. Cheng, „Ode-inspired network design for single image super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1732–1741 (cit. on p. 42).

[159] N. Ahn, B. Kang, and K.-A. Sohn, „Fast, accurate, and lightweight super-resolution with cascading residual network," in *European Conference on Computer Vision*, 2018, pp. 252–268 (cit. on pp. 43, 44).

[160] X. Chu, B. Zhang, H. Ma, R. Xu, and Q. Li, „Fast, accurate and lightweight super-resolution with neural architecture search," in *International Conference on Pattern Recognition*, 2020, pp. 59–64 (cit. on pp. 43, 44).

[161] W. Li, K. Zhou, L. Qi, N. Jiang, J. Lu, and J. Jia, „Lapar: Linearly-assembled pixel-adaptive regression network for single image super-resolution and beyond," *arXiv preprint arXiv:2105.10422*, 2021 (cit. on pp. 43, 44).

[162] X. Luo, Y. Xie, Y. Zhang, Y. Qu, C. Li, and Y. Fu, „Latticenet: Towards lightweight image super-resolution with lattice block," in *European Conference on Computer Vision*, 2020, pp. 272–289 (cit. on pp. 43, 44).

[163] S. Gu, L. Zhang, W. Zuo, and X. Feng, „Weighted nuclear norm minimization with application to image denoising," in *IEEE conference on computer vision and pattern recognition*, 2014, pp. 2862–2869 (cit. on pp. 45, 46).

[164] D. Martin, C. Fowlkes, D. Tal, and J. Malik, „A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *IEEE International Conference on Computer Vision*, 2001, pp. 416–423 (cit. on pp. 46, 47).

[165] Z. Xia and A. Chakrabarti, „Identifying recurring patterns with deep neural networks for natural image denoising," in *IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 2426–2434 (cit. on p. 47).

[166] C. Tian, Y. Xu, and W. Zuo, „Image denoising using deep cnn with batch renormalization," *Neural Networks*, vol. 121, pp. 461–473, 2020 (cit. on p. 47).

[167] R. Franzen, „Kodak lossless true color image suite," *source: http://rok. us/graphics/kodak*, vol. 4, no. 2, 1999 (cit. on p. 47).

[168] L. Zhang, X. Wu, A. Buades, and X. Li, „Color demosaicking by local directional interpolation and nonlocal adaptive thresholding," *Journal of Electronic imaging*, vol. 20, no. 2, p. 023 016, 2011 (cit. on p. 47).

[169] A. Foi, V. Katkovnik, and K. Egiazarian, „Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images," *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1395–1411, 2007 (cit. on p. 48).

[170] H. Sheikh, „Live image quality assessment database release 2," *http://live. ece. utexas. edu/research/quality*, 2005 (cit. on p. 48).

[171] W. Li, X. Tao, T. Guo, L. Qi, J. Lu, and J. Jia, „Mucan: Multi-correspondence aggregation network for video super-resolution," in *European Conference on Computer Vision*, 2020, pp. 335–351 (cit. on pp. 54, 56, 68, 69, 82, 84, 94).

[172] Y. Tian, Y. Zhang, Y. Fu, and C. Xu, „Tdan: Temporally-deformable alignment network for video super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3360–3369 (cit. on pp. 54, 56, 82–84, 99).

[173] D. Fuoli, S. Gu, and R. Timofte, „Efficient video super-resolution through recurrent latent space propagation," in *IEEE International Conference on Computer Vision Workshop*, 2019, pp. 3476–3485 (cit. on pp. 54, 56, 68, 69, 82, 84, 94).

[174] Y. Huang, W. Wang, and L. Wang, „Bidirectional recurrent convolutional networks for multi-frame super-resolution," *Advances in Neural Information Processing Systems*, vol. 28, pp. 235–243, 2015 (cit. on pp. 54, 56, 82, 84, 88).

[175] T. Isobe, X. Jia, S. Gu, S. Li, S. Wang, and Q. Tian, „Video super-resolution with recurrent structure-detail network," in *European Conference on Computer Vision*, 2020, pp. 645–660 (cit. on pp. 54, 56, 68, 69, 82, 84, 94).

[176] K. C. Chan, X. Wang, K. Yu, C. Dong, and C. C. Loy, „Basicvsr: The search for essential components in video super-resolution and beyond," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4947–4956 (cit. on pp. 54, 56, 62, 67–70, 82–84, 87, 93–95).

[177] K. C. Chan, S. Zhou, X. Xu, and C. C. Loy, „Basicvsr++: Improving video super-resolution with enhanced propagation and alignment," *arXiv preprint arXiv:2104.13371*, 2021 (cit. on pp. 54, 56, 57, 62, 64, 66, 68–70, 81–84, 87–89, 91, 93–95).

[178] J. Caballero, C. Ledig, A. Aitken, A. Acosta, J. Totz, Z. Wang, and W. Shi, „Real-time video super-resolution with spatio-temporal networks and motion compensation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4778–4787 (cit. on pp. 54, 56, 82, 84).

[179] Y. Huang, W. Wang, and L. Wang, „Video super-resolution via bidirectional recurrent convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 1015–1028, 2017 (cit. on pp. 54, 56, 82, 84).

[180] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang, „Deep video deblurring for hand-held cameras," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1279–1288 (cit. on pp. 54, 56, 66, 67, 72–75, 82, 84, 92, 93, 95, 97).

[181] S. Zhou, J. Zhang, J. Pan, H. Xie, W. Zuo, and J. Ren, „Spatio-temporal filter adaptive network for video deblurring," in *IEEE International Conference on Computer Vision*, 2019, pp. 2482–2491 (cit. on pp. 54, 56, 72–74, 82, 84, 97).

[182] T. Isobe, S. Li, X. Jia, S. Yuan, G. Slabaugh, C. Xu, Y.-L. Li, S. Wang, and Q. Tian, „Video super-resolution with temporal group attention," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8008–8017 (cit. on pp. 54, 56, 57, 68, 69, 82, 84, 94).

[183] D. Li, C. Xu, K. Zhang, X. Yu, Y. Zhong, W. Ren, H. Suominen, and H. Li, „Arvo: Learning all-range volumetric correspondence for video deblurring," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7721–7731 (cit. on pp. 54, 56, 67, 72–74, 82, 84, 93, 97).

[184] S. Nah, S. Son, and K. M. Lee, „Recurrent neural networks with intra-frame iterations for video deblurring," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8102–8111 (cit. on pp. 54, 56, 72, 73, 82, 84, 97).

[185] Z. Zhong, Y. Gao, Y. Zheng, and B. Zheng, „Efficient spatio-temporal recurrent neural network for video deblurring," in *European Conference on Computer Vision*, 2020, pp. 191–207 (cit. on pp. 54, 56, 73, 82, 84, 97).

[186] H. Son, J. Lee, J. Lee, S. Cho, and S. Lee, „Recurrent video deblurring with blur-invariant motion estimation and pixel volumes," *ACM Transactions on Graphics*, vol. 40, no. 5, pp. 1–18, 2021 (cit. on pp. 54, 56, 72–74, 82, 84, 97).

[187] M. S. Sajjadi, R. Vemulapalli, and M. Brown, „Frame-recurrent video super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6626–6634 (cit. on pp. 54, 56, 68, 69, 82–84, 87, 94).

[188] M. Haris, G. Shakhnarovich, and N. Ukita, „Recurrent back-projection network for video super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3897–3906 (cit. on pp. 54, 56, 69, 77).

[189] T. Isobe, F. Zhu, X. Jia, and S. Wang, „Revisiting temporal modeling for video super-resolution," *arXiv preprint arXiv:2008.05765*, 2020 (cit. on pp. 54, 56, 68, 69, 82, 84, 94).

[190] J. Lin, Y. Huang, and L. Wang, „Fdan: Flow-guided deformable alignment network for video super-resolution," *arXiv preprint arXiv:2105.05640*, 2021 (cit. on pp. 54, 56, 68, 69, 82, 84, 94).

[191] A. Greaves-Tunnell and Z. Harchaoui, „A statistical investigation of long memory in language and music," in *International Conference on Machine Learning*, 2019, pp. 2394–2403 (cit. on p. 55).

[192] L. Wang, Y. Wang, Z. Liang, Z. Lin, J. Yang, W. An, and Y. Guo, „Learning parallax attention for stereo image super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 250–12 259 (cit. on p. 56).

[193] Y. Guo, J. Chen, J. Wang, Q. Chen, J. Cao, Z. Deng, Y. Xu, and M. Tan, „Closed-loop matters: Dual regression networks for single image super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5407–5416 (cit. on p. 56).

[194] Y. Fan, J. Yu, D. Liu, and T. S. Huang, „Scale-wise convolution for image restoration," in *AAAI Conference on Artificial Intelligence*, 2020, pp. 10 770–10 777 (cit. on p. 56).

[195] X. Xiang, Q. Lin, and J. P. Allebach, „Boosting high-level vision with joint compression artifacts reduction and super-resolution," in *International Conference on Pattern Recognition*, 2021, pp. 2390–2397 (cit. on p. 56).

[196] L. Wang, Y. Wang, Z. Lin, J. Yang, W. An, and Y. Guo, „Learning a single network for scale-arbitrary super-resolution," in *IEEE Conference on International Conference on Computer Vision*, 2021, pp. 10 581–10 590 (cit. on p. 56).

[197] J. Li, Z. Pei, and T. Zeng, „From beginner to master: A survey for deep learning-based single-image super-resolution," *arXiv preprint arXiv:2109.14335*, 2021 (cit. on p. 56).

[198] L. Sun, C. Sakaridis, J. Liang, Q. Jiang, K. Yang, P. Sun, Y. Ye, K. Wang, and L. Van Gool, „Mefnet: Multi-scale event fusion network for motion deblurring," *arXiv preprint arXiv:2112.00167*, 2021 (cit. on pp. 56, 84).

[199] Y. Fan, H. Shi, J. Yu, D. Liu, W. Han, H. Yu, Z. Wang, X. Wang, and T. S. Huang, „Balanced two-stage residual networks for image super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 161–168 (cit. on p. 56).

[200] J. Yu, Y. Fan, J. Yang, N. Xu, Z. Wang, X. Wang, and T. Huang, „Wide activation for efficient and accurate image super-resolution," *arXiv preprint arXiv:1808.08718*, 2018 (cit. on p. 56).

[201] D. Liu, Z. Wang, Y. Fan, X. Liu, Z. Wang, S. Chang, X. Wang, and T. S. Huang, „Learning temporal dynamics for video super-resolution: A deep learning approach," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3432–3445, 2018 (cit. on p. 56).

[202] P. Yi, Z. Wang, K. Jiang, J. Jiang, T. Lu, X. Tian, and J. Ma, „Omniscient video super-resolution," in *IEEE International Conference on Computer Vision*, 2021, pp. 4429–4438 (cit. on pp. 56, 68, 69, 94).

[203] M. Maggioni, Y. Huang, C. Li, S. Xiao, Z. Fu, and F. Song, „Efficient multi-stage video denoising with recurrent spatio-temporal fusion," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3466–3475 (cit. on p. 56).

[204] G. Vaksman, M. Elad, and P. Milanfar, „Patch craft: Video denoising by deep modeling and patch matching," in *IEEE International Conference on Computer Vision*, 2021, pp. 1759–1768 (cit. on pp. 56, 67, 75, 76, 98).

[205] S. Lee, D. Cho, J. Kim, and T. H. Kim, „Restore from restored: Video restoration with pseudo clean video," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3537–3546 (cit. on p. 56).

[206] R. Yang and R. Timofte, „Ntire 2021 challenge on quality enhancement of compressed video: Methods and results," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 647–666 (cit. on p. 56).

[207] P. Yi, Z. Wang, K. Jiang, J. Jiang, and J. Ma, „Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations," in *IEEE International Conference on Computer Vision*, 2019, pp. 3106–3115 (cit. on pp. 56, 66–69, 93, 94).

[208] X. Xiang, H. Wei, and J. Pan, „Deep video deblurring using sharpness features from exemplars," *IEEE Transactions on Image Processing*, vol. 29, pp. 8976–8987, 2020 (cit. on pp. 56, 72, 73, 97).

[209] X. Xiang, Y. Tian, Y. Zhang, Y. Fu, J. P. Allebach, and C. Xu, „Zooming slow-mo: Fast and accurate one-stage space-time video super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3370–3379 (cit. on pp. 56, 70, 77, 84).

[210] J. Pan, H. Bai, and J. Tang, „Cascaded deep video deblurring using temporal sharpness prior," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3043–3051 (cit. on pp. 56, 72–74, 97).

[211] L. Wang, Y. Guo, L. Liu, Z. Lin, X. Deng, and W. An, „Deep video super-resolution using hr optical flow estimation," *IEEE Transactions on Image Processing*, vol. 29, pp. 4323–4336, 2020 (cit. on pp. 56, 84).

[212] D. Y. Sheth, S. Mohan, J. L. Vincent, R. Manzorro, P. A. Crozier, M. M. Khapra, E. P. Simoncelli, and C. Fernandez-Granda, „Unsupervised deep video denoising," in *IEEE International Con-*

*ference on Computer Vision*, 2021, pp. 1759–1768 (cit. on pp. 56, 84).

[213] S. Li, F. He, B. Du, L. Zhang, Y. Xu, and D. Tao, „Fast spatio-temporal residual network for video super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 522–10 531 (cit. on p. 56).

[214] R. Liao, X. Tao, R. Li, Z. Ma, and J. Jia, „Video super-resolution via deep draft-ensemble learning," in *IEEE International Conference on Computer Vision*, 2015, pp. 531–539 (cit. on pp. 56, 84).

[215] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, „Video enhancement with task-oriented flow," *International Journal of Computer Vision*, vol. 127, no. 8, pp. 1106–1125, 2019 (cit. on pp. 56, 62, 66–70, 75–77, 82–84, 92–94, 99).

[216] K. C. Chan, X. Wang, K. Yu, C. Dong, and C. C. Loy, „Understanding deformable alignment in video super-resolution," in *AAAI Conference on Artificial Intelligence*, 2021, pp. 973–981 (cit. on pp. 56, 83).

[217] A. Kappeler, S. Yoo, Q. Dai, and A. K. Katsaggelos, „Video super-resolution with convolutional neural networks," *IEEE Transactions on Computational Imaging*, vol. 2, no. 2, pp. 109–122, 2016 (cit. on pp. 56, 84).

[218] D. Liu, Z. Wang, Y. Fan, X. Liu, Z. Wang, S. Chang, and T. Huang, „Robust video super-resolution with learned temporal dynamics," in *IEEE International Conference on Computer Vision*, 2017, pp. 2507–2515 (cit. on pp. 56, 57, 84).

[219] X. Tao, H. Gao, R. Liao, J. Wang, and J. Jia, „Detail-revealing deep video super-resolution," in *IEEE International Conference on Computer Vision*, 2017, pp. 4472–4480 (cit. on pp. 56, 84).

[220] A. Ranjan and M. J. Black, „Optical flow estimation using a spatial pyramid network," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4161–4170 (cit. on pp. 56, 61, 65, 82, 84, 90–92).

[221] Y. Jo, S. W. Oh, J. Kang, and S. J. Kim, „Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3224–3232 (cit. on pp. 56, 68, 69, 84, 94).

[222] T. H. Kim, M. S. Sajjadi, M. Hirsch, and B. Scholkopf, „Spatio-temporal transformer network for video restoration," in *European Conference on Computer Vision*, 2018, pp. 106–122 (cit. on pp. 56, 73, 97).

[223] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, „Deformable convolutional networks," in *IEEE International Conference on Computer Vision*, 2017, pp. 764–773 (cit. on pp. 56, 64, 82–85, 92).

[224] N. Shazeer, „Glu variants improve transformer," *arXiv preprint arXiv:2002.05202*, 2020 (cit. on pp. 57, 65).

[225] X. Li, Y. Hou, P. Wang, Z. Gao, M. Xu, and W. Li, „Trear: Transformer-based rgb-d egocentric action recognition," *IEEE Transactions on Cognitive and Developmental Systems*, 2021 (cit. on pp. 57, 60).

[226] C. Wick, J. Zöllner, and T. Grüning, „Transformer for handwritten text recognition using bidirectional post-decoding," in *International Conference on Document Analysis and Recognition*, 2021, pp. 112–126 (cit. on pp. 57, 60).

[227] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, „Deep learning for 3d point clouds: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 12, pp. 4338–4364, 2021 (cit. on p. 57).

[228] G. Bertasius, H. Wang, and L. Torresani, „Is space-time attention all you need for video understanding?" *arXiv preprint arXiv:2102.05095*, 2021 (cit. on p. 57).

[229] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, „Vivit: A video vision transformer," *arXiv preprint arXiv:2103.15691*, 2021 (cit. on p. 57).

[230] D. Neimark, O. Bar, M. Zohar, and D. Asselmann, „Video transformer network," *arXiv preprint arXiv:2102.00719*, 2021 (cit. on p. 57).

[231] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, „Video swin transformer," *arXiv preprint arXiv:2106.13230*, 2021 (cit. on pp. 57, 62).

[232] O. Ronneberger, P. Fischer, and T. Brox, „U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241 (cit. on p. 59).

[233] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, „Flownet: Learning optical flow with convolutional networks," in *IEEE International Conference on Computer Vision*, 2015, pp. 2758–2766 (cit. on pp. 61, 84, 89).

[234] S. Niklaus, *A reimplementation of SPyNet using PyTorch*, https://github.com/sniklaus/pytorch-spynet, 2018 (cit. on pp. 61, 65, 89, 92).

[235] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, „Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8934–8943 (cit. on pp. 61, 82, 84, 89).

[236] I. Loshchilov and F. Hutter, „Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016 (cit. on pp. 65, 92).

[237] S. Nah, S. Baik, S. Hong, G. Moon, S. Son, R. Timofte, and K. Mu Lee, „Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 1996–2005 (cit. on pp. 66, 68–71, 73, 74, 92–95, 98).

[238] C. Liu and D. Sun, „On bayesian adaptive video super resolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 2, pp. 346–360, 2013 (cit. on pp. 66, 67, 69, 70, 76, 77, 93–95).

[239] A. Khoreva, A. Rohrbach, and B. Schiele, „Video object segmentation with language referring expressions," in *Asian Conference on Computer Vision*, 2018, pp. 123–141 (cit. on pp. 66, 67, 74–76, 92, 93, 97, 98).

[240] M. Haris, G. Shakhnarovich, and N. Ukita, „Recurrent back-projection network for video super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3897–3906 (cit. on pp. 68, 82, 84, 94).

[241] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, „Scale-recurrent network for deep image deblurring," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8174–8182 (cit. on pp. 72–74, 97).

[242] K. Zhang, W. Luo, Y. Zhong, L. Ma, W. Liu, and H. Li, „Adversarial spatio-temporal learning for video deblurring," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 291–301, 2018 (cit. on pp. 72, 73).

[243] M. Suin, K. Purohit, and A. Rajagopalan, „Spatially-attentive patch-hierarchical network for adaptive motion deblurring," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3606–3615 (cit. on pp. 73, 74).

[244] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, „Multi-stage progressive image restoration," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 821–14 831 (cit. on pp. 73, 74, 97).

[245] P. Arias and J.-M. Morel, „Video denoising via empirical bayesian estimation of space-time patches," *Journal of Mathematical Imaging and Vision*, vol. 60, no. 1, pp. 70–93, 2018 (cit. on pp. 75, 76, 98).

[246] X. Xu, L. Siyao, W. Sun, Q. Yin, and M.-H. Yang, „Quadratic video interpolation," *Advances in Neural Information Processing Systems*, vol. 32, 2019 (cit. on pp. 75, 76).

[247] T. Kalluri, D. Pathak, M. Chandraker, and D. Tran, „Flavr: Flow-agnostic video representations for fast frame interpolation," *arXiv preprint arXiv:2012.08512*, 2020 (cit. on pp. 75, 76).

[248] K. Soomro, A. R. Zamir, and M. Shah, „Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012 (cit. on pp. 75, 76).

[249] W. Bao, W.-S. Lai, C. Ma, X. Zhang, Z. Gao, and M.-H. Yang, „Depth-aware video frame interpolation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3703–3712 (cit. on pp. 76, 77).

[250] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala, „Video frame synthesis using deep voxel flow," in *IEEE International Conference on Computer Vision*, 2017, pp. 4463–4471 (cit. on p. 76).

[251] M. Choi, H. Kim, B. Han, N. Xu, and K. M. Lee, „Channel attention is all you need for video frame interpolation," in *AAAI Conference on Artificial Intelligence*, 2020, pp. 10 663–10 671 (cit. on p. 76).

[252] J. Park, K. Ko, C. Lee, and C.-S. Kim, „Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation," in *European Conference on Computer Vision*, 2020, pp. 109–125 (cit. on p. 76).

[253] H. Lee, T. Kim, T.-y. Chung, D. Pak, Y. Ban, and S. Lee, „Adacof: Adaptive collaboration of flows for video frame interpolation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5316–5325 (cit. on p. 76).

[254] M. Haris, G. Shakhnarovich, and N. Ukita, „Space-time-aware multi-resolution video enhancement," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2859–2868 (cit. on p. 77).

[255] G. Xu, J. Xu, Z. Li, L. Wang, X. Sun, and M.-M. Cheng, „Temporal modulation network for controllable space-time video super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6388–6397 (cit. on p. 77).

[256] Z. Geng, L. Liang, T. Ding, and I. Zharkov, „Rstt: Real-time spatial temporal transformer for space-time video super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 441–17 451 (cit. on pp. 77, 85).

[257] Y. Zhang, Y. Zhang, Y. Wu, Y. Tao, K. Bian, P. Zhou, L. Song, and H. Tuo, „Improving quality of experience by adaptive video streaming with super-resolution," in *IEEE Conference on Computer Communications*, 2020, pp. 1957–1966 (cit. on p. 81).

[258] H. Liu, Z. Ruan, P. Zhao, C. Dong, F. Shang, Y. Liu, L. Yang, and R. Timofte, „Video super-resolution based on deep learning: A comprehensive survey," *Artificial Intelligence Review*, pp. 1–55, 2022 (cit. on p. 81).

[259] Z. Wan, B. Zhang, D. Chen, and J. Liao, „Bringing old films back to life," *arXiv preprint arXiv:2203.17276*, 2022 (cit. on pp. 81, 89).

[260] M. Chu, Y. Xie, J. Mayer, L. Leal-Taixé, and N. Thuerey, „Learning temporal coherence via self-supervision for gan-based video generation," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 75–1, 2020 (cit. on p. 82).

[261] X. Zhu, H. Hu, S. Lin, and J. Dai, „Deformable convnets v2: More deformable, better results," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9308–9316 (cit. on pp. 83–85, 92).

[262] J. Lin, Y. Cai, X. Hu, H. Wang, Y. Yan, X. Zou, H. Ding, Y. Zhang, R. Timofte, and L. Van Gool, „Flow-guided sparse transformer for video deblurring," *arXiv preprint arXiv:2201.01893*, 2022 (cit. on pp. 84, 85, 89, 97).

[263] J. Caballero, C. Ledig, A. Aitken, A. Acosta, J. Totz, Z. Wang, and W. Shi, „Real-time video super-resolution with spatio-temporal networks and motion compensation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4778–4787 (cit. on p. 84).

[264] Z. Tu, H. Talebi, H. Zhang, F. Yang, P. Milanfar, A. Bovik, and Y. Li, „Maxvit: Multi-axis vision transformer," *arXiv preprint arXiv:2204.01697*, 2022 (cit. on p. 85).

[265] Z. Xia, X. Pan, S. Song, L. E. Li, and G. Huang, „Vision transformer with deformable attention," *arXiv preprint arXiv:2201.00520*, 2022 (cit. on p. 85).

[266] X. Dong, J. Bao, D. Chen, W. Zhang, N. Yu, L. Yuan, D. Chen, and B. Guo, „Cswin transformer: A general vision transformer backbone with cross-shaped windows," *arXiv preprint arXiv:2107.00652*, 2021 (cit. on p. 85).

[267] W. Yun, M. Qi, C. Wang, H. Fu, and H. Ma, „Coarse-to-fine video denoising with dual-stage spatial-channel transformer," *arXiv preprint arXiv:2205.00214*, 2022 (cit. on p. 85).

[268] C. Liu, H. Yang, J. Fu, and X. Qian, „Learning trajectory-aware transformer for video super-resolution," *arXiv preprint arXiv:2204.04216*, 2022 (cit. on p. 85).

[269] Z. Tu, H. Talebi, H. Zhang, F. Yang, P. Milanfar, A. Bovik, and Y. Li, „Maxim: Multi-axis mlp for image processing," *arXiv preprint arXiv:2201.02973*, 2022 (cit. on pp. 85, 97).

[270] D. Fuoli, M. Danelljan, R. Timofte, and L. Van Gool, „Fast online video super-resolution with deformable attention pyramid," *arXiv preprint arXiv:2202.01731*, 2022 (cit. on p. 85).

[271] M. Cao, Y. Fan, Y. Zhang, J. Wang, and Y. Yang, „Vdtr: Video deblurring with transformer," *arXiv preprint arXiv:2204.08023*, 2022 (cit. on p. 85).

[272] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, „Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020 (cit. on p. 85).

[273] Z. Wang, Y. Ma, Z. Liu, and J. Tang, „R-transformer: Recurrent neural network enhanced transformer," *arXiv preprint arXiv:1907.05572*, 2019 (cit. on p. 89).

[274] J. Lei, L. Wang, Y. Shen, D. Yu, T. L. Berg, and M. Bansal, „Mart: Memory-augmented recurrent transformer for coherent video paragraph captioning," *arXiv preprint arXiv:2005.05402*, 2020 (cit. on p. 89).

[275] J. Gu and C. Dong, „Interpreting super-resolution networks with local attribution maps," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9199–9208 (cit. on p. 105).

[276] L. Xie, X. Wang, C. Dong, Z. Qi, and Y. Shan, „Finding discriminative filters for specific degradations in blind super-resolution," *Advances in Neural Information Processing Systems*, vol. 34, pp. 51–61, 2021 (cit. on p. 105).

[277] Y. Liu, A. Liu, J. Gu, Z. Zhang, W. Wu, Y. Qiao, and C. Dong, „Discovering distinctive" semantics" in super-resolution networks," *arXiv preprint arXiv:2108.00406*, 2021 (cit. on p. 105).

[278] K. Zhang, W. Zuo, and L. Zhang, „Deep plug-and-play super-resolution for arbitrary blur kernels," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1671–1681 (cit. on p. 105).

[279] J. Song, C. Meng, and S. Ermon, „Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020 (cit. on p. 105).

[280] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, „High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695 (cit. on p. 105).

# NOTATION

## PRACTICAL DEGRADATION MODEL

| SYMBOL | MEANING |
|---|---|
| $\mathbf{y}$ | low-resolution image |
| $\mathbf{x}$ | high-resolution image |
| $\mathbf{k}$ | Gaussian blur kernel |
| $\downarrow_{\mathbf{s}}$ | downsampling operation with scale factor $\mathbf{s}$ |
| $\mathbf{n}$ | white Gaussian noise with standard deviation $\sigma$ |
| $\mathbf{B}_{\text{iso}}$ | Gaussian blur operation with isotropic Gaussian kernel |
| $\mathbf{B}_{\text{aniso}}$ | Gaussian blur operation with anisotropic Gaussian |
| $\mathbf{D}^{\mathbf{s}}_{\text{nearest}}$ | nearest downsampling |
| $\mathbf{D}^{\mathbf{s}}_{\text{bilinear}}$ | bilinear downsampling |
| $\mathbf{D}^{\mathbf{s}}_{\text{bicubic}}$ | bicubic downsampling |
| $\mathbf{D}^{\mathbf{s}}_{\text{down-up}}$ | equals to $\mathbf{D}^{\mathbf{s}/\mathbf{a}}_{\text{down}}\mathbf{D}^{\mathbf{a}}_{\text{up}}$), first downsamples the image with a scale factor $\mathbf{s}/\mathbf{a}$ and then upscales with a scale factor $\mathbf{a}$ |
| $\mathbf{N}_{\text{G}}$ | Gaussian noise |
| $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ | three-dimensional zero-mean Gaussian noise with covariance matrix $\mathbf{\Sigma}$ |
| $\mathbf{N}_{\text{JPEG}}$ | JPEG image compression artifacts/noise |
| $\mathbf{N}_{\text{S}}$ | processed camera sensor noise |

IMAGE RESTORATION

| SYMBOL | MEANING |
|---|---|
| $I_{LQ}$ | low-quality image |
| $I_{HQ}$ | high-quality image |
| $H_{SF}(\cdot)$ | the shallow feature extraction module |
| $H_{RSTB_i}(\cdot)$ | the $i$-th RSTB block |
| $F_i$ | intermediate feature of the $i$-th layer |
| $H_{DF}(\cdot)$ | the deep feature extraction module |
| $F_DF$ | deep feature |
| $I_{RHQ}$ | reconstructed high-quality image |
| $\epsilon$ | a constant in the Charbonnier loss |
| $H_{STL_{i,j}}(\cdot)$ | the $j$-th Swin Transformer layer in the $i$-th RSTB block |
| $P_Q$ | the projection matrix for query |
| $P_K$ | the projection matrix for key |
| $P_V$ | the projection matrix for value |
| $Q$ | the query matrix |
| $K$ | the key matrix |
| $V$ | the value matrix |
| $B$ | the learnable relative positional encoding |
| $d$ | the channel number of query |
| $M$ | attention window size |
| SoftMax | the softmax operation along the column direction |
| MSA | the multi-head self-attention |
| MLP | the multi-layer perceptron |
| LN | the layerNorm layer |

VIDEO RESTORATION

| SYMBOL | MEANING |
|---|---|
| $I^{LQ}$ | a sequence of low-quality frames |
| $I^{HQ}$ | a sequence of high-quality frames |
| $I^{SF}$ | a sequence of shallow frame features |
| $I^{DF}$ | a sequence of deep frame features |
| $I^{RHQ}$ | a sequence of reconstructed high-quality frames |
| $X^R$ | reference frame feature |
| $X^S$ | supporting frame feature |
| $P^Q$ | the projection matrix for query |
| $P^K$ | the projection matrix for key |
| $P^V$ | the projection matrix for value |
| $Q^R$ | the query matrix from the reference frame |
| $K^S$ | the key matrix from the supporting frame |
| $V^S$ | the value matrix from the supporting frame |
| $D$ | the channel number of projected feature |
| $M$ | attention window size |
| $\mathrm{MA}(\cdot)$ | the mutual attention |
| $A$ | correlation matrix |
| $Y^R_{i,:}$ | the refined feature of the $i$-th element in the reference frame |
| MMA | the multi-head mutual attention |
| $X_t$ | the $t$-th frame feature |
| $\hat{X}_{t-1}$ | the warped $(t-1)$-th frame towards the $t$ frame |
| $O_{t-1,t}$ | the optical flow from the $(t-1)$-th frame to the $t$-th frame |
| $\mathcal{W}$ | the image warping function |
| $X'_{t-1}$ | the initial warped feature of the $(t-1)$-th frame |
| $o_{t-1,t}$ | the offset residual from the $(t-1)$-th frame to the $t$-th frame |
| $m_{t-1,t}$ | the modulation mask for the optical flow from the $(t-1)$-th frame to the $t$-th frame |
| $\mathcal{D}$ | the deformable convolution |
| $\hat{X}_{t-1}$ | the final aligned feature of the $(t-1)$-th frame |
| $F^i_t$ | the $t$-th clip feature at the $i$-th layer |
| $\widehat{F}^i_{t-1}$ | the $(t-1)$-th aligned clip feature towards the $t$-th clip at the $i$-th layer |

## COLOPHON

This document was typeset in LaTeX using the typographical look-and-feel `classicthesis`. The bibliography is typeset using `biblatex`.