GDAŃSK UNIVERSITY OF TECHNOLOGY

The author of the doctoral dissertation: Piotr Kopa Ostrowski
Scientific discipline: Automation, Electronics, Electrical Engineering and Space Technologies

DOCTORAL DISSERTATION

Title of doctoral dissertation: Efficient Deep Neural Networks for Video Denoising

Title of doctoral dissertation (in Polish): Wydajne głębokie sieci neuronowe do usuwania szumów w sygnale wideo

Supervisor *signature*
dr hab. inż. Tomasz Stefański

Gdańsk, year 2025

Summary

Video denoising constitutes a fundamental component of video processing pipelines, with applications ranging from mobile video calls and augmented reality to autonomous systems and medical imaging. It is particularly important in low-light environments and when the size of the capturing device has to be minimised. Over the past decade, artificial neural networks have revolutionised most areas of computer image processing, including video denoising. Despite this progress, achieving high-quality restoration under efficiency-constrained conditions, such as real-time processing and low latency, remains a challenge. In lightweight neural network models, there is a risk of excessive reliance on the reference frame, leading to sub-optimal utilisation of temporal context. This dissertation investigates techniques to improve the quality of efficient deep video denoisers through enhanced exploitation of temporal information. Two complementary strategies are proposed.

The first strategy involves pre-training neural network models under conditions in which the signal from the denoised frame is restricted. This encourages implicit motion estimation and stronger reliance on neighbouring frames. Two implementations of this approach are presented, i.e., the denoised frame is masked during the pre-training stage, or noise in the denoised frame is amplified. To further facilitate the use of temporal context, the noise in neighbouring frames is reduced during pre-training. As these techniques are applied exclusively during training, they do not affect inference time, resulting in an improved balance between quality and efficiency.

The second strategy focuses on explicit motion estimation, which may be necessary for small models when significant motion occurs. This approach employs the optical flow estimated from heavily downsized frames. Reducing the resolution of the inputs to the optical flow estimator substantially lowers computational cost, while the resulting flow maps still provide useful motion information. The resulting coarse alignment enables significant improvements in denoising quality without strongly affecting the efficiency.

Both strategies are formally described in the dissertation, and their individual and combined effects are assessed through extensive experiments. The positive impact of the strategies is demonstrated both quantitatively and qualitatively. Ablation studies validate design decisions and highlight the complementary nature of the proposed methods. The results show that the proposed techniques substantially improve video denoising performance across a range of datasets and scenarios, including

sequences with severe motion and with real or very strong noise. Comparisons of the model combining both strategies with state-of-the-art methods indicate that it is competitive or superior. Deployment of this model on multiple edge devices proves its practical applicability.

Finally, the dissertation identifies open challenges and outlines potential directions for future research, including applications of the presented solutions to other video modalities and further advances in efficient temporal processing. The proposed methods may also be utilised by practitioners developing efficient video restoration pipelines for diverse applications.

Streszczenie

Odszumianie stanowi fundamentalny element procesów przetwarzania wideo, znajdując zastosowanie w tak różnorodnych obszarach jak rozmowy wideo, wirtualna i rozszerzona rzeczywistość, systemy autonomiczne czy obrazowanie medyczne. Szczególne znaczenie ma ono w warunkach słabego oświetlenia oraz w sytuacjach, gdy rozmiar urządzenia rejestrującego musi zostać zminimalizowany. W ciągu ostatniej dekady sztuczne sieci neuronowe zrewolucjonizowały większość obszarów komputerowego przetwarzania obrazu, w tym również odszumianie wideo. Pomimo tego postępu, uzyskanie wysokiej jakości rekonstrukcji przy jednoczesnym zachowaniu wysokiej efektywności, tj. przetwarzaniu w czasie rzeczywistym lub przy niskiej latencji, wciąż stanowi wyzwanie. W przypadku małych modeli sieci neuronowych istnieje ryzyko nadmiernego polegania na odszumianej klatce wideo, co prowadzi do nieoptymalnego wykorzystania kontekstu czasowego. Niniejsza rozprawa bada techniki poprawy jakości modeli uczenia maszynowego do odszumiania wideo poprzez lepsze wykorzystanie informacji z sąsiednich klatek. Zaproponowane zostały dwie uzupełniające się strategie.

Pierwsza strategia opiera się na wstępnym uczeniu modeli sieci neuronowych w warunkach, w których sygnał z odszumianej klatki jest ograniczony. Sprzyja to lepszemu oszacowaniu ruchu przez model, który nie posiada do tego wyspecjalizowanych mechanizmów oraz silniejszemu wykorzystaniu klatek sąsiednich. Przedstawiono dwie implementacje tego podejścia. W pierwszej z nich odszumiana klatka jest maskowana podczas wstępnego etapu uczenia, w drugiej szum w klatce odszumianej jest wzmacniany. Aby dodatkowo ułatwić wykorzystanie kontekstu czasowego, w trakcie wstępnego uczenia poziom szumu w klatkach sąsiednich jest ograniczany. W związku z tym, że techniki te stosowane są wyłącznie na etapie treningu, nie wpływają one na czas przetwarzania podczas inferencji, skutkując modelami o poprawionym stosunku jakości do efektywności.

Druga strategia koncentruje się na bezpośrednim oszacowaniu ruchu, które może być niezbędne dla małych modeli w sytuacjach występowania silnych ruchów pomiędzy klatkami wideo. Podejście to wykorzystuje przepływ optyczny estymowany na znacząco pomniejszonych klatkach. Redukcja rozdzielczości wejść estymatora przepływu optycznego znacząco obniża koszty obliczeniowe, przy czym uzyskane mapy nadal dostarczają użytecznych informacji o ruchu. Uzyskane w ten sposób zgrubne dopasowanie sąsiednich ramek wideo umożliwia istotną poprawę jakości

odszumiania wideo bez znaczącego obniżenia efektywności.

Obie strategie zostały formalnie opisane w rozprawie, a ich indywidualne oraz łączne efekty dokładnie przedstawiono w eksperymentach. Ich pozytywny wpływ wykazano zarówno ilościowo, jak i jakościowo. Przeprowadzone badania potwierdzają słuszność podjętych decyzji dotyczących szczegółów obu strategii i podkreślają komplementarność zaproponowanych podejść. Wyniki pokazują, że proponowane metody znacząco poprawiają jakość odszumiania wideo w różnych zbiorach danych i scenariuszach, w tym w sekwencjach z silnym ruchem oraz z rzeczywistym bądź bardzo intensywnym szumem. Porównania opracowanego modelu łączącego obie strategie do odszumiania wideo w formacie RAW z dotychczasowymi metodami wskazują, że przedstawione podejście, w zależności od zbioru danych, jest konkurencyjne lub lepsze. Implementacja zaproponowanego modelu na wielu urządzeniach brzegowych potwierdza jego praktyczną użyteczność.

Wreszcie, rozprawa identyfikuje wciąż istniejące wyzwania oraz wskazuje potencjalne kierunki dalszych badań, w tym zastosowanie zaproponowanych metod do innych typów wideo i dalszy rozwój wydajnych metod wykorzystujących kontekst czasowy. Zaproponowane metody mogą również zostać wykorzystane przez praktyków rozwijających wydajne systemy przywracania jakości wideo do różnorodnych zastosowań.

Acknowledgements

I would like to thank all my supervisors and colleagues I met along my doctoral studies, from whom I learned and with whom I had the opportunity to work. I am also immensely grateful to Łucja, my family, and friends who supported me throughout this difficult journey and believed, often more than I did, that it would end in success.

List of abbreviations

The following abbreviations has been used throughout this thesis.

CNN	Convolutional Neural Network
GPU	Graphics Processing Unit
FIP	Frame Interpolation Pre-training
NI	Noise Imbalancing
LVD	Lightweight Video Denoiser
RNN	Recurrent Neural Network
CRVD	Captured Raw Video Denoising
ReCRVD	Recaptured Raw Video Denoising Dataset
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index Measure
LPIPS	Learned Perceptual Image Patch Similarity
W(E)	Warping Error
FPS	Frames per Second
GMACs	Giga Multiply–Accumulate Operations
SFD	Single Frame Difference
PCIF	Percentage of Correct Intrinsic Flow
LROF	Low-Resolution Optical Flow
x	corrupted, noisy frame
y	clean frame
$\mathcal{N}$	Gaussian distribution
σ	standard deviation of the Gaussian noise
$\mathcal{P}$	Poisson distribution
α	scaling factor of the Poisson distribution
b_g	background value of the Poisson distribution
$\Pi_{[L,U]}$	clipping to the range $< L, U >$
x_t	corrupted, noisy frame at time step t
y_t	clean frame at time step t
$\tilde{y}_t$	restored frame frame at time step t
X_t	set of input frames with x_t and its neighbours
m	noise map
N	neural network
$\mathcal{M}_\Theta$	tree-like architecture with multiple networks

D	depth of the $\mathcal{M}_\Theta$
d	depth level of the $\mathcal{M}_\Theta$
$\mathcal{B}_\tau^d$	block of the $\mathcal{M}_\Theta$ at depth d with center frame for time step t
N^d	neural network employed at depth d of the $\mathcal{M}_\Theta$
y_τ^d	output of the $\mathcal{B}_\tau^d$
i_τ^d	input of the $\mathcal{B}_\tau^d$
$\mathcal{L}_{\text{den}}$	loss function utilized during denoising
$\mathcal{L}_{\text{int}}$	loss function utilized during FIP stage
e	epoch number
E	number of training epochs
E_{FIP}	number of training epochs employing FIP
$\hat{x}_t$	noisy frame at time step t with reduced noise
$\mathcal{G}_1$	gate function controlling the noise reduction factor
$\mathbf{0}$	tensor of the same shape as the input frame filled with zeros
$\overline{\mathcal{B}}_\tau^D$	modified block of the $\mathcal{M}_\Theta$ used during Post-FIP stage at depth D with center frame for time step t
$\mathcal{G}_2$	gate function controlling the scaling factor for the residual connection
it	iteration number during Post-FIP stage
I	number of iterations in Post-FIP stage
b	bias
l	pixel location
g_{t-k}	gradient map computed for frame x_{t-k} with respect to $\tilde{y}_t$
$u_{t\pm k,l}$	intrinsic flow between frames x_t and $x_{t\pm k}$ at location l
$v_{t\pm k,l}$	optical flow between frames x_t and $x_{t\pm k}$ at location l
$r1$, $r2$, $r3$	PCIF thresholds
Loc	set of locations chosen for PCIF computation
$X_t^{\mathcal{W}}$	set of input frames with x_t and neighbours warped with respect to x_t
$\mathcal{W}(x_{i1}, x_{i2})$	frame x_{i1} warped with respect to frame x_i2
$OF(x_{i1}, x_{i2})$	optical flow between frame x_{i2} and x_{i1}
s	scaling factor employed for computing LROF
$\downarrow_s$	downsizing by the factor s
$\uparrow_s$	upsampling by the factor s
h	pre-processing function applied to inputs of the optical flow estimators
$\overline{OF}$	filtered optical flow, with vectors of magnitude below the threshold set to zero
ρ	threshold for filtering optical flow
b_f	size of the Bayer filter
X'	set of input frames following Bayer pattern reshaped with pixel unshuffle
$\overline{X}_t$	set of input frames after NI
Y_t	set of clean frames with y_t and its neighbours
f	factor used to scale noise for the NI
F	initial value of the factor used to scale noise for the NI

Contents

Chapter 1

Introduction

1.1 Foreword and Motivation

Denoising is one of the oldest and most extensively studied tasks in image processing. It can be defined as the process of removing random, unwanted variations from a corrupted signal. Noise arising in images can come from multiple sources: imperfections in the capturing device (e.g., patchy distribution of silver halide crystals in analogue photography or non-homogeneous response of pixels in digital camera sensors), the discrete nature of light (photon shot noise), or external factors such as the impact of temperature and voltage fluctuations in digital cameras.

The primary goal of denoising has traditionally been to enhance the plausibility and legibility of visual content for human observers. However, with the growing number of computer vision applications, improving the quality of images and videos for subsequent processing in high-level tasks has become increasingly important. Finally, denoising models are found to be useful in two non-obvious applications. In the first case, denoisers are employed as priors to regularise optimisation for a wide set of inverse problems, i.a., deblurring, super-resolution, inpainting. The second field is image generation, where diffusion-based models outperform other image generating techniques like generative adversarial networks and variational autoencoders. Diffusion models are trained to reverse the process of progressive noising and, as a result during inference, they are able to generate image from noise.

The period between 1980 and 2010 was marked by the development of multiple approaches aimed at effective image denoising. These methods relied on various techniques such as Bayesian optimisation, filtering, non-local averaging, and processing in the transform domain. Progress, however, began to stagnate towards the end of the first decade of the 21st century. Since various approaches produced similar results and classical methods were no longer able to yield significant improvements, some researchers questioned whether the peak of image denoising capabilities had already been reached [1, 2, 3].

A few years later, AlexNet [4] revolutionised computer vision by not only win-

ning the ImageNet Large Scale Visual Recognition Challenge [5], but also by outperforming all the competing approaches by a large margin. While neural networks, and specifically convolutional neural networks (CNNs), had been introduced much earlier (e.g., Neocognitron in 1980 [6] and LeNet in 1989 [7]), their impact remained limited. Apart from algorithmic and architectural advancements, two other factors were essential for their breakthrough: the availability of large-scale training datasets, exemplified by the ImageNet challenge data, and substantial computational power required for training deep neural networks made possible through the emergence of graphics processing units (GPUs). GPUs are specialised hardware designed to perform a large number of similar operations in parallel, which turned out to be necessary for large-scale deep learning. With the increasing depth of neural networks and the introduction of fundamental architectural innovations such as residual connections and normalisation layers, the accuracy of deep learning models increased rapidly. These advances were soon extended to a wide range of tasks, initially within computer vision, including segmentation [8, 9] and detection [10, 11], but later also to other domains. Progress in natural language processing led to incredible development of Large Language Models like GPT-3 [12] or Llama [13]. The combination of deep and reinforcement learning enabled performance beyond the human level in multiple board and video games [14, 15, 16, 17], and later was successfully applied in robotics [18] and autonomous vehicles [19]. Significant advances were also reached in speech recognition [20, 21], speech synthesis [22, 23, 24], recommender systems [25, 26], cybersecurity [27, 28, 29], and even drug discovery [30, 31].

Inevitably deep learning was also adopted for image and, later, video denoising. Since the publication of DnCNN [32], neural networks have dominated the field, achieving performance levels unreachable for classical methods. Following this remarkable success, the field of denoising continues to develop dynamically. Large-scale models yield increasingly impressive results [33, 34, 35], new training paradigms enable learning even without access to clean reference data [36, 37, 38, 39], thus narrowing the gap between supervised and self-supervised approaches. Furthermore, recent advances improve the models ability to generalise to wide range of noise types, also these unseen during training [40, 41]. Despite significant progress in video denoising, achieving outstanding denoising results under efficiency-constraint conditions remains a challenge. State-of-the-art methods usually rely on complex architectures, which hinder their deployment in many practical scenarios.

The popularity of video telephony grew rapidly with the proliferation of smartphones and the introduction of video calling into messaging applications. It has accelerated further since the COVID-19 pandemic, which made working from home very popular and thus led to a sudden increase in video conferencing. Denoising is a crucial element of RAW data processing, especially for small cameras in mobile phones and laptops, the most popular devices for video calls. Signal latency above 150 ms is known to be a distracting factor [42] thus it is important to minimise it, including through the use of fast denoising algorithms.

Live video streaming has also seen a surge in popularity in recent years, thereby

increasing the demand for real-time video denoising. It is especially necessary in applications where streaming video is used to control the recorded objects or the capturing device itself. Examples in the field of medicine are laparoscopy, endoscopy, camera- and microscopy-enhanced dentistry, and telemedicine. Advances in denoising algorithms enable the miniaturisation of video capture devices, thereby increasing surgical safety and patient comfort. Real-time processing and low latency are also essential for remotely controlled vehicles. Important examples include underwater remotely operated vehicles and non-autonomous unmanned aerial vehicles, which are often used in the conditions when recording high-quality videos is difficult. Moreover, autonomous robots, cars and drones commonly utilise camera information for navigation and decision-making, consequently, a high-quality, low-latency signal is required.

The market booming in recent years is that of video surveillance. A hampering factor can be the high initial investment largely driven by the cost of high-quality CCTV systems. These costs could be reduced by using lower-quality cameras if the frames quality would be subsequently restored with algorithmic methods. To truly reduce the costs, processing should not introduce significant additional overhead, which can be avoided by performing restoration locally on budgetary processing units. Such an approach would require efficient, but well-performing video denoising methods, especially for low-light scenarios when the impact of noise is the most critical.

Real-time and low-latency processing is also crucial for augmented and virtual reality. The use of augmented and virtual reality systems can cause motion sickness in some users, with latency being one of the primary causes [43]. Therefore, minimising the temporal lookahead required to process or further restore a frame is essential. Moreover, latency reduces the sense of immersion, thereby downgrading the overall quality of the user experience. As denoising constitutes a fundamental component of Image Signal Processing and video restoration pipelines, improving its efficiency directly contributes to enhanced user experience. Furthermore, denoising is employed in virtual reality to accelerate scene rendering, particularly in ray tracing, where it serves to reduce the sampling noise and enable faster rendering of photorealistic scenes.

Another example demonstrating the importance of denoising concerns the use of machine learning models trained for high-level vision tasks, e.g., classification, segmentation, and detection. These models tend to perform best on data following the distribution represented in the training set. A model evaluated on inputs degraded in ways not represented during training may exhibit substantially lower performance compared to that on clean data. [44]. Thus restoring quality, among others by performing denoising, prior to applying high-level tasks can be crucial to ensure the optimal performance of machine learning models.

Beyond the previously discussed applications that require efficient video denoising, lightweight and fast algorithms solving this task are also highly desirable

in a broader range of scenarios as they reduce computational costs, improve user experience, and enable their deployment on a wider variety of devices. In particular, deployment on edge devices, which is usually constrained to lightweight models, offers several substantial benefits over cloud computing. Specifically, it reduces bandwidth consumption, minimises latency introduced by data transfer, and enhances privacy by limiting the need to offload data to external servers. In a similar way, the implementation on battery-powered devices requires the use of energy-efficient models to prevent excessive reduction in their operating time. All these factors, combined with the growing popularity of ever-increasing video resolutions, underscore the critical importance of efficient video denoising.

1.2 Scope and Contributions

The goal of this dissertation is to present techniques developed in order to improve the quality of efficient deep video denoisers through better temporal context utilisation. Twofold, complementary strategies are proposed: training that enforces implicit motion estimation and lightweight explicit alignment. In this work, emphasis is placed on videos recorded in the visible spectrum, i.e., preprocessed RGB or RAW data as they are the most popular and involve the broadest range of applications. Nevertheless, the proposed techniques could potentially be extended to other modalities like hyperspectral, ultrasound, depth or event-based videos.

The widespread use of video denoising has fuelled the development of numerous methods designed to efficiently address this task. However, most of them lack mechanisms specifically designed to leverage the temporal context. For efficiency reasons, some approaches are restricted to concatenating video frames or features extracted during processing [45, 46]. Others propose architectural mechanisms designed to better exploit neighbouring frames, but these are limited, i.e., they are restricted to static regions of an image [47] or expand the receptive field in a blind manner [48]. A popular strategy relies on concurrent denoising of multiple frames [49, 50] or even entire sequences [34, 51]. While this approach can improve the frames-per-second–quality trade-off of the models, it inevitably increases latency, which may be prohibitive in certain applications. The use of optical flow computed on downsized images has never been thoroughly investigated. Previous attempts were limited to moderate downsizing of the input images [51] and the precise impact of such downsizing has not been systematically studied. The training aspect is also underexplored as very few training techniques dedicated specifically to video denoising have been proposed to date. To alleviate flickering between frames, *Overlap Loss* is proposed [52] and frame-interpolation is incorporated in the training of MVDenoiser [53] to take advantage of pre-training on a vast amount of videos.

First hypothesis: Limiting access to information from the reference frame during the initial stage of the training can lead to the improvement

of the temporal context utilisation and further enhance quality of video denoising.

The assumption is that a model with restricted access to information from the denoised (reference) frame will be encouraged to estimate the clean frame by relying heavily on neighbouring frames. Although the outputs of a model pre-trained under such conditions may be of lower quality than those of a model with the same architecture trained in the standard way, it is expected that the pre-trained model will develop a stronger ability to exploit temporal context. In the second phase of the training, when the model is optimised for the target denoising task, it can preserve this enhanced ability to leverage temporal information while simultaneously learning less difficult utilisation of the information contained in the reference frame. This two-stage training can help mitigate the models tendency to focus excessively on the central frame, which constitutes a possible shortcut in video denoising, especially for lightweight models. Two methods that implement the described training strategy are investigated:

- **Frame Interpolation Pre-training (FIP)** in which, during the initial epochs, the model is trained on a proxy task, i.e., frame interpolation,

- **Noise Imbalancing (NI)** in which, during the initial epochs, the noise in the reference frame is magnified and in other frames it is reduced.

Second hypothesis: Effective explicit motion estimation can be achieved by computing optical flow on radically downsized input frames. Leveraging this flow is expected to provide significant improvements in video denoising without harming the computational efficiency.

Implicit motion estimation allows a model to exploit information from neighbouring frames even when the motion between these frames and the reference frame occurs. However, this capability is inherently limited. It is theoretically bounded by the receptive field, but in practice, especially for lightweight networks, the effective range is often much smaller. As stated in the first hypothesis, it is assumed that this limitation can be alleviated, for instance, through appropriate pre-training, but implicit motion estimation alone is expected to be insufficient to handle severe motion.

Optical flow has been widely adopted in a wide range of video restoration tasks and architectures; however, its application has generally been restricted to the methods that do not prioritise computational efficiency. To overcome this constraint, the dissertation investigates the use of optical flow estimated from frames downsized by factors of up to 12. This downsizing drastically reduces the computational cost of the flow estimation. Although the flow derived from low-resolution frames cannot provide highly precise warping, it can still effectively co-align pixels from neighbouring frames, while finer alignment can subsequently be handled implicitly by the network.

1.3 Dissertation outline

The remaining part of this dissertation is structured as follows. Chapter 2 presents relevant prior research that provides the background to efficient neural network processing and video denoising. It begins with efficiency-related topics, including architectural innovations, methods for model compression, and strategies for more effective training. In the second part, video denoising is discussed, starting with a broad introduction to the restoration domain. It is followed by a review of video restoration techniques. Afterwards, potential sources of noise are discussed, and for both image and video denoising the available datasets, classical approaches, and deep learning methods are presented. The chapter concludes with identification of literature gaps that this dissertation seeks to address.

Chapter 3 presents FIP, the method using frame interpolation as a proxy task to pre-train video denoising models. Firstly, a formal description of the proposed pre-training and the subsequent Post-FIP stage is introduced. In an ablation study, the details of the proposed method are investigated qualitatively and quantitatively. The comparison between baseline training and training augmented with FIP across multiple architectures demonstrates that employing frame interpolation as a proxy pre-training task improves the utilisation of neighbouring frames and leads to higher overall video denoising quality. Finally, the results of the models utilising the proposed pre-training are compared with state-of-the-art methods thus demonstrating that they are either competitive or superior in certain scenarios.

Chapter 4 shows that using optical flow derived from low-resolution frames enables significant improvements in denoising quality, specifically for datasets with severe motion. For the data containing mainly a static background with individual moving objects, it is shown that the proposed pre-training technique, i.e., NI is more beneficial. The combination of both techniques is also presented. It gives the best results across the considered datasets proving that the proposed techniques are complementary. For both methods formal descriptions are presented. In an ablation study, design decisions relating to the methods details are validated. The resulting model incorporating both proposed techniques, referred to as the **Lightweight Video Denoiser (LVD)**, is compared with state-of-the-art methods for efficient video denoising. LVD achieves superior results on the dataset featuring real-motion and delivers results comparable to the state-of-the-art methods on dataset with predominantly static scenes. Finally, the deployment results across multiple platforms are demonstrated, highlighting the practical applicability of LVD.

Finally, Chapter 5 discusses the contributions presented throughout this dissertation. In addition, it outlines several directions for future research, including the application of the proposed techniques to new tasks and the exploration of potential extensions of these methods.

Chapter 2

Background and related works

Since the release of AlexNet in 2012 [4], deep learning has significantly transformed the field of artificial intelligence. By leveraging large datasets and GPU-accelerated computations, it has revolutionised multiple domains, beginning with computer vision tasks such as image classification, object detection, tracking, and later extending to natural language processing, with conversational agents becoming its most prominent example. Deep learning has also impacted decision-making systems, including those used for playing board and video games.

Beyond these well-known applications, deep learning has also brought about substantial, though less widely recognised, advancements in the areas of image restoration and enhancement and, more recently, their extension to video processing. One of the most appealing restoration tasks is denoising, which is particularly critical in scenarios such as low-light environments, medical imaging, and scientific imaging, where noise can severely degrade the visual quality and hinder subsequent analysis. With the increasing popularity of deep learning in real-world systems, the need for efficient processing, both in terms of computation time and resource utilisation, has become critical. This is particularly relevant for scenarios requiring real-time performance on resource-constrained edge devices, where achieving high-quality results without compromising efficiency remains a significant challenge.

In this chapter, the aspects of deep learning most relevant to this dissertation are discussed. Section 2.1 focuses on neural network efficiency, beginning with an overview of architectural approaches to building efficient models (Section 2.1.1), followed by model optimisation techniques (Section 2.1.2), and concluding with training strategies that improve the model performance without sacrificing its efficiency (Section 2.1.3). Section 2.2 presents the state of the art in image and video restoration, beginning with general image restoration methods, followed by advancements in video restoration (Section 2.2.1), and culminating in a focused review of denoising methods for both images and video (Sections 2.2.2 and 2.2.3 respectively). Finally, Section 2.3 summarises the identified gaps in the current literature that this dissertation aims to address.

2.1 Neural Network Efficiency

Due to the continuously increasing size of neural network models and even faster expansion of their applications, performance has become a critical issue. Efficient models are necessary not only to minimise latency but also to reduce the already substantial computational costs of training and inference. Moreover, the increasing deployment of such models on edge devices imposes additional constraints as these devices typically possess significantly lower computational capabilities than standard GPUs, yet are still often expected to deliver real-time or near-real-time performance.

2.1.1 Efficient Architectures

The foundation of an efficient deep learning models lies in their architecture. In the context of computer vision, several architectural innovations have been introduced to reduce models computational complexity without compromising the performance.

One widely adopted strategy involves replacing standard convolutions with their more efficient alternatives. Depthwise separable convolutions, as used in various MobileNet [54, 55, 56] variants, significantly reduce the number of operations by factorising a standard convolution into a depthwise one followed by a pointwise convolution. MobileNetV2 [55] extends this idea by introducing an inverted residual structure with linear bottlenecks, where depthwise layers are sandwiched between pointwise convolutions that reduce and then restore the channel dimensions. This design creates a compact bottleneck enabling efficient computation. The authors of SqueezeNet [57] propose replacing a significant proportion of the popular 3×3 kernels with the simplest 1×1 kernels thus demonstrating that applying spatial operations universally can be redundant. Convolution can be further factorised by replacing standard square kernels $n \times n$ with a stack of 1-dimensional kernels, i.e., $1 \times n$ and $n\times1$, like in [58, 59, 60]. Another effective technique is the use of group convolutions, where subsets of input channels are processed independently to reduce the number of computations. Initially used in AlexNet [4], this concept has been refined to maintain high accuracy while enabling faster execution. An example of architecture focused on reducing convolutional overhead by using group convolution is ShuffleNet [61], which adds a shuffling layer between two group convolution operations, enabling connectivity between all the input and output channels of the module. The idea of processing channels independently is extended in ResNeXt [62], where the channels are divided into chunks, processed independently through multiple layers, and finally the results from all the chunks are summed up.

To reuse the already computed features, DenseNet [63] introduces the idea of dense connections between the layers, i.e., each layer uses the feature maps of the preceding layers as inputs. As a result, subsequent layers can focus on exploiting new features and still take advantage of the already computed ones. Another approach is

proposed in GhostNet [64], where instead of utilising a large number of kernels only a small proportion of them is used and the missing feature maps are created, based on individual channels, from the kernel results. To improve the models quality without harming efficiency, multiple lightweight attention mechanisms are introduced. Modules such as Squeeze-and-Excitation [65], Efficient Channel Attention [66] or Convolutional Block Attention Module [67] aggregate values for feature maps across spatial dimensions using operations such as max pooling or average pooling, process them with simple operations, and use them as factors to selectively emphasise informative features. With the widespread adoption of the self-attention mechanism in computer vision and other domains, the need for more efficient variants has become critical. The standard self-attention formulation exhibits quadratic complexity with respect to the number of tokens, which becomes prohibitive for high-resolution inputs. Consequently, a variety of approaches are proposed to reduce this complexity while retaining most of the representational power of global attention. They include local window attention [33, 68], axial attention [69], and low-rank approximations such as Nyström attention [70].

To avoid heavy computations, an early exit mechanism is introduced. Taking advantage of auxiliary classifiers has already been introduced in Inception net [71], but in this approach auxiliary classifiers are removed during inference. In the networks with the early exit mechanism, the final result can be produced by the intermediate layers of a network for easy samples. Early exits are especially beneficial for problems with varying complexity across the cases. While it can lead to significant average speed-ups, it cannot provide a predictable efficiency improvement. Similarly, not the entire model is utilised in the approaches related to the Mixture of Experts [72] strategy. This kind of model contains multiple subnetworks or layers called experts. During inference only some of them, chosen dynamically based on the input, are activated. In an extreme case, only one expert is chosen, which minimises the processing costs. This approach is used to solve multiple problems, among others super-resolution [73]. While dynamic processing provides significant theoretical acceleration, it can be hard to achieve in certain scenarios because deep learning hardware and software are often optimized for processing static models [74].

Finally, to improve efficiency, certain components of neural networks can be simplified or removed during inference without altering the model outputs. A prominent example is the batch normalisation layer. During training, this layer normalises the output of a convolutional or fully connected layer using batch-wise statistics (mean and variance computed from the current mini-batch). However, during inference, these dynamic statistics are replaced with fixed values, i.e, running estimates of the mean and variance accumulated throughout the training. This transformation converts the batch normalisation layer into a deterministic linear transformation, which can be fused with the preceding layer if this layer is linear. This fusion reduces computational overhead and leads to more efficient inference without impacting the model behavior. Similarly, consecutive or parallel convolutional layers can be merged into a single equivalent layer at inference time, provided there are no intervening non-

linearities such as activation functions. This process, often referred to as structural reparameterisation, enables the simplification of network architectures without altering their functional behavior. Several studies have demonstrated that training a network with additional layers or branches, designed to be reparameterised into a more compact structure during deployment, can lead to better performance than training a smaller model from scratch [75, 76, 77]. This improvement is largely attributed to more favorable optimisation dynamics and gradient flow during training.

A range of specialised, task-specific architectural solutions have been introduced to address particular computer vision tasks. In classification, it is common to reduce the spatial size of the processed input in the initial layers. In segmentation, early downsizing is also a popular choice; the final result is then upscaled using standard, non-trainable operations to the original resolution. Processing data at a lower scale enables taking wider context into account at the price of losing some details. Some models, such as ICNet [78], apply multi-resolution processing by extracting features at different scales and fusing them before upsampling. Bilateral networks (e.g., BiSeNet [79]) adopt a two-pathway design, with a shallow branch for spatial details and a deep branch for the semantic context, later fused via attention mechanisms. Fast-SCNN [80] builds upon this design by introducing a shared and efficient downsizing module across both paths, minimising the redundancy. Other dual-resolution models, such as DDRNet [81], incorporate feature sharing and bidirectional communication between the branches to enhance accuracy. For efficient context modeling, they often employ pyramid pooling or multi-kernel modules to blend their features across scales. For tasks that require dense outputs, such as segmentation, optical flow estimation and various restoration tasks, it is common to employ architectures that process images on multiple scales. In the encoder of such a model, the spatial resolution of activation maps is gradually reduced, while in the decoder, the original resolution is progressively restored. Such an approach enables an increase in the receptive field and facilitates gathering of information even from more distant pixels. The examples of such an approach include SegNet [82], U-Net [83], Enet [84], ERFNet [59], ESPNet [85]. An important insight from the authors of Enet is that the size of the encoder can be significantly smaller than the size of the decoder, as its main role is merely to fine-tune the details. Among detection architectures, SSD [86] achieves a good trade-off between accuracy and efficiency by resigning from heavy modules for the region proposal, and predicting bounding boxes directly from backbone features. For the image enhancement task, some authors decide to use lookup tables encoding relationships between the observed and the expected colours, instead of dense pixel-value predictions which are more complex. To make the process even simpler, a group of base lookup tables can be precomputed, and the model can estimate only the weights used to aggregate them [87, 88].

Beyond manual design, Neural Architecture Search offers a data-driven approach to discovering efficient architectures tailored to specific performance targets or hardware platforms. By evaluating a wide variety of architectural configurations, including different layer types, connections, and activation functions, Neural Archi-

tecture Search can identify optimal solutions for multiple tasks, i.a. image classification [89], segmentation [90], super-resolution [91], and denoising [92]. Although Neural Architecture Search can be computationally expensive due to the large search space, techniques have been developed to reduce its cost, including proxy-based [93] and differentiable [94, 95] methods. Notably, results show that optimal architectures can vary depending on the target device. For instance, shallower networks with early resolution reduction and large kernels may perform better on GPUs, while deeper architectures with smaller kernels are preferable for central processing units [94].

2.1.2 Model Compression

Beyond the previously discussed layer fusion techniques, several other methods have been proposed to accelerate deep neural networks during inference. These approaches aim to reduce the computational cost and memory footprint of the trained models, often with minimal or no loss in accuracy. One of the most prominent examples is quantisation. The central idea is to represent network parameters and activations using lower-precision numerical formats, i.e., typically 8-bit integers instead of 32-bit floating-point values used during training. It enables faster arithmetic operations and reduces memory bandwidth requirements, especially on edge devices and specialised accelerators. While 8-bit arithmetic is the most widely supported among integer representations, numerous attempts are made to utilise even smaller precisions (such as 4-bit [96, 97] or even ternary [98] and binary [99] networks), although maintaining high quality results becomes more difficult in these scenarios. Therefore, models designed to operate in lower precisions are not simply quantised after training (Post-Training Quantisation), but are instead trained in a manner that mitigates the limitations of low-bit representations (Quantisation-Aware Training). On the other hand, the 16-bit floating point representation (half precision) can be considered a default approach for deployment, as it can usually be applied with minimal effort, significant acceleration, and a negligible drop in accuracy. Not only inference but also training can be accelerated with half precision [100]. Modern frameworks support a range of quantisation schemes, including uniform vs. non-uniform quantisation, per-layer vs. per-channel scaling, and symmetric vs. asymmetric quantisation, each offering trade-offs between implementation simplicity, efficiency, and accuracy preservation. When carefully applied, quantisation often achieves significant latency and energy improvements with minimal loss in model performance.

The second widely adopted approach to improving inference efficiency is pruning. Similar to reparameterisation techniques, pruning is based on the insight that it is often easier to train a large model with abundant parameters and then reduce its complexity rather than train a compact model from scratch. Pruning removes redundant or less important parameters to reduce the computational cost and memory usage. This process can target various levels of the network, including individual weights, channels, or even entire layers and blocks. However, the choice of pruning granularity is often constrained by the deployment platform as many hardware and

software environments cannot effectively leverage unstructured sparsity. Among the available strategies, channel pruning [101, 102] is the most commonly used due to its favourable balance between parameter reduction and minimal accuracy degradation, while also enabling straightforward and practical acceleration on standard hardware. According to the Lottery Ticket Hypothesis [103], pruning a model can reveal a sub-network which, when trained from scratch using the same initial weights, can match the accuracy of the original full model.

For specific hardware–software configurations, additional efficiency gains can be achieved by exploiting sparsity in neural networks. Sparsity can manifest itself in two primary forms, i.e., as weight sparsity referring to zero-valued model parameters, and as activation sparsity referring to zero outputs during inference. Weight sparsity is often encouraged through regularisation techniques such as ℓ_1 regularisation or weight decay, which promote the elimination of less important parameters. Activation sparsity naturally arises in networks employing the ReLU activation function, which sets all negative values to zero thus delivering significant proportion of zero activations. In both cases, further sparsity can be introduced by thresholding small-magnitude values to zero thereby enhancing the potential for computational savings. Sparsity can also be enlarged naturally during quantisation as quantisation schemes are typically designed to ensure that zero is one of the representable values in the quantised domain. However, the actual benefits of sparsity depend heavily on the capability of the underlying hardware and inference engine of exploiting sparsity patterns efficiently.

Other explored directions in the context of model compression are weight clustering and low-rank approximation. Weight clustering [104, 105] relies on replacing parameters with similar values by a single value. While it enables the reduction of the model size, it does not directly lead to the model acceleration. The second approach enables reducing the number of operations during inference by replacing large tensors of weights with their approximation using, e.g., Tucker decomposition [106] or CP-decomposition [107].

2.1.3 Training Efficient Models

Optimisation techniques such as quantisation, pruning, and sparsification typically degrade the model performance when applied the neural network training. To mitigate this drawback, models can be trained in ways that minimise performance loss.

For quantisation, a commonly used approach is the previously mentioned Quantisation-Aware Training. Although all the training operations are conducted with floating-point precision, the forward pass simulates the quantisation of weights and activations in accordance with the target inference precision. Since the quantisation process involves non-differentiable operations (particularly rounding) the Straight-Through Estimator [108] is used during the backward pass. Straight-Through Estimator approximates gradients as if no quantisation was applied, which allows for

the use of standard optimisation techniques such as stochastic gradient descent.

For pruning, it is common to fine-tune the model after the parameter reduction. To achieve better results, pruning can be performed incrementally: a small proportion of the parameters is removed and the model is fine-tuned repeatedly. This gradual pruning strategy often yields superior performance compared to a single large pruning step [109, 110]. A major challenge in pruning consists in determining which parameters to remove. One common strategy is based on heuristics, e.g., treating weights with the smallest magnitudes as the least important. Although intuitive and easy to implement, this method often lacks accuracy. To achieve better pruning performance, more sophisticated techniques, which rely on Hessian information, have been proposed in order to more accurately estimate the impact of weight removal on the loss function [111, 112]. However, for large networks, computing the full Hessian matrix is computationally infeasible. To address this limitation, more efficient approximations such as K-FAC have been introduced [113]. Alternatively, sparsity can be encouraged directly during training thus enabling straightforward pruning by removing weights or channels that converge to zero. Regularisation techniques such as the ℓ_0, ℓ_1, ℓ_2 penalties have been widely explored in order to induce sparsity in neural networks [114, 115, 116]. However, since these methods typically lead to unstructured sparsity, they do not effectively translate into acceleration on most hardware platforms. To address this limitation, methods specifically designed to enforce group sparsity have been developed [117, 114]. They enable structured pruning that aligns better with hardware constraints and allows for more efficient inference.

Since large neural network models are capable of capturing complex patterns within data, they have been proposed as an additional source of supervision. This concept is known as knowledge distillation. Originally proposed for the classification problem [118], this approach involves training a smaller model (student) using soft labels generated by a larger, pre-trained model (teacher) instead of the conventional one-hot ground truth labels. These soft labels encode information about class similarities, allowing the training loss to penalise ambiguous errors less strongly than obvious ones. This idea has since been extended to incorporate intermediate representations rather than relying solely on the final outputs. In such settings, the teacher's features serve as an additional supervisory signal rather than a complete replacement for standard supervision. However, directly matching activation maps between the teacher and the student is often impractical due to the differences in the layer dimensions or network architecture. To address this issue, several improvements have been introduced. One strategy involves using a projection layer to transform student features before comparing them to the teacher's features [119], enabling more compatible representations. Other methods use attention maps derived from the teacher's features as the supervisory signal, avoiding the need for direct feature alignment [120], or rely on preserving the similarity between the training examples [121]. Among many other fields, knowledge distillation has also been successfully applied to video restoration tasks [122, 123, 124].

In addition to the efficiency-oriented training techniques mentioned above, general training improvements play a crucial role in developing high-performing models. These improvements allow for better performance to be achieved under a fixed computational budget. In restoration tasks, the selection of an appropriate loss function is especially important. The most commonly used loss functions include the ℓ_1, ℓ_2, and SSIM-based losses, but to produce more visually plausible results, they are often supplemented with additional terms. For instance, perceptual loss [125] compares frames in the feature space rather than in the colour space. While features are typically extracted from the early layers of a pre-trained network, it has also been shown that networks with randomly initialised weights can still provide meaningful feature representations for loss computation [126]. Another approach that enhances realism is adversarial loss applied via the Generative Adversarial Networks [127] framework. However, this method often trades fidelity for realism, which can be undesirable in certain applications. Additional losses have also been proposed to enforce specific desired properties such as:

- sharpness [128, 129],
- smoothness [130],
- temporal consistency [131, 132, 133, 52],
- colour preservation [134].

Another commonly used strategy to improve neural network training is data augmentation, which aims to enrich the training dataset by applying various transformations to the input-target pairs. However, achieving high-quality results requires careful selection of these transformations, as even simple augmentations can negatively affect the performance, particularly in low-level tasks such as denoising. For example, in RAW image denoising, operations such as the rotation, cropping, or flipping can disrupt the Bayer pattern. Similarly, upscaling and downsizing may introduce spatial correlations within otherwise uncorrelated real noise, while colour or lighting adjustments can artificially change the noise variance. To preserve the Bayer pattern during training, task-specific geometric transformations have been proposed [135]. For video restoration, temporal augmentations such as frame order reversal and subsampling have been shown to be beneficial [136]. Models trained on specific noise distributions often tend to overfit, leading to poor generalisation on unseen noise types. To mitigate this, a masking-based augmentation strategy is proposed [40]. This method requires not only input data augmentation but also a specific model architecture to be effective. In contrast, a model-independent approach known as Adversarial Frequency Mixup has also been proposed [41]. This technique augment dataset by mixing a noisy frame with its denoised counterpart in the frequency domain with the aim of exposing the model to a wider range of noise characteristics. To determine the optimal frequency components for mixing, a small auxiliary network is trained as part of the process. In the context of Noise2Noise training [36], where ground-truth clean images are unavailable, a self-augmentation strategy is proposed

to synthetically generate suitable training pairs [38]. For microscopy data, a differentiable augmentation approach is explored to adaptively learn augmentations during training [137]. A closely related area is noise modeling, which enables the generation of noise following distributions observed in real-world datasets and thereafter allows leveraging clean, noiseless data for effective model training [138, 139, 140, 141].

A popular approach in deep learning is fine-tuning models pre-trained on large datasets to solve specific problems. This is a standard practice in high-level vision tasks such as segmentation, classification, or detection. The intuition behind this is that filters, especially in the earlier layers, learn to recognise generally useful visual patterns such as edges, shapes, and colours. In contrast, this approach is less common in image restoration tasks. These tasks typically require different types of features that may not be shared across domains, and large-scale datasets containing clean/noisy image pairs are more difficult to obtain. Nevertheless, several exceptions exist. For example, one study explores leveraging pre-trained image denoisers to improve video denoising performance [142]. The authors of Captured Raw Video Denoising dataset (CRVD) [143], one of the first video datasets featuring real noise, propose initially training the model on a larger dataset corrupted with Poisson-Gaussian noise, which approximately matches the characteristics of real-world noise from CRVD. Recently, the authors of MVDenoiser [53] have proposed using masked central-frame modeling to pre-train their model on massive available ordinary videos.

Similarly, self-supervised techniques have been proposed to pre-train models on proxy tasks, helping to build useful prior knowledge before training on the target restoration task. For high-level vision problems, common proxy tasks include rotation prediction [144], colourisation [145], and masked image modelling [146, 147]. For video data, additional tasks such as next-frame prediction [148] or temporal order prediction [149] are also used.

2.2 Restoration

In the domain of computer vision, restoration refers to a class of tasks concerned with reconstructing a high-quality signal (image or video) from a degraded or corrupted input. Beyond denoising, the most prominent tasks within this category include super-resolution, motion deblurring, defocus deblurring, demosaicking, inpainting, colourisation, rain/snow/fog removal, and compression artefact removal. Closely related to restoration is the task of image enhancement, which focuses on improving the visual plausibility or perceptual quality of an image. A notable example is low-light enhancement, which involves brightening poorly illuminated regions while preserving image details and avoiding noise amplification. Another related task is frame interpolation, which aims to synthesise intermediate frames between the existing ones in a video sequence. This task is particularly valuable for increasing frame rates, generating smooth slow-motion effects, and enhancing temporal continuity. Frame interpolation methods typically rely on estimating motion (e.g., optical

flow) or learning temporal correspondences, with the aim of accurately predicting the content of the missing frames, while maintaining spatial coherence and temporal consistency.

Restoration algorithms play a crucial role in modern computer vision systems by enabling high-quality image or video reconstruction from degraded inputs, thereby significantly extending the capabilities of imaging hardware. This is especially important in applications where physical constraints limit either the quality or the size of sensors, e.g., in mobile devices, medical imaging equipment, and robotic platforms. In mobile phones, for instance, the demand for compact, lightweight, and power-efficient designs necessitates the use of small sensors and lenses, which inherently struggle with noise, limited dynamic range, and optical aberrations. Restoration algorithms, such as denoising, deblurring, and super-resolution, can bridge this gap by computationally increasing the quality of an image after it has been captured. Similarly, in medical imaging modalities, such as endoscopy or capsule cameras, the sensor size must be minimised to ensure patient safety and comfort, at the same time making sophisticated restoration techniques essential for acquiring diagnostically useful imagery. Beyond standard RGB imaging, restoration is equally critical in other medical imaging modalities such as X-ray, computed tomography, and magnetic resonance imaging, where denoising techniques are widely used to reduce the effects of quantum noise or acquisition-related artefacts while preserving anatomical details. In some cases, deblurring methods are also used, e.g., in computed tomography or magnetic resonance imaging scans distorted by patient movement. Restoration in these contexts enables clearer interpretation by radiologists and improves the performance of automated diagnostic tools. Furthermore, in optical and fluorescence microscopy, where capturing high-resolution, high-SNR images is often constrained by phototoxicity, limited exposure times, and diffraction limits, denoising is essential to reveal underlying biological structures. In more advanced settings, techniques such as deblurring and super-resolution reconstruction are also applied to enhance spatial resolution and clarity beyond the physical limitations of imaging systems [150, 151, 152, 153]. These restoration methods support accurate analysis in biomedical research and clinical diagnostics, where image fidelity is crucial for its correct interpretation and for decision making. In robotics and autonomous systems, limited payload and power constraints often require the use of lightweight cameras that produce lower-quality images, particularly in challenging environments. Moreover, restoration algorithms can reduce costs, which is crucial in many practical applications. By allowing the use of lower-grade optics or sensors and still achieving acceptable or even high-quality outputs, they eliminate the need for expensive hardware upgrades. This is especially beneficial in large-scale deployments such as surveillance systems, industrial inspection pipelines, or consumer electronics, where minimising the unit cost is a key design requirement. Moreover, restoration algorithms are indispensable in overcoming difficult environmental scenarios such as low-light or high-dynamic-range scenes, fast-moving objects that induce motion blur, atmospheric distortions in outdoor settings (e.g., haze, rain) or underwater environ-

ments where scattering and colour shifts occur. By compensating for physical and environmental limitations of image acquisition, restoration algorithms not only improve visual fidelity but also enhance the performance of downstream tasks such as object detection, recognition and tracking, thereby enabling robust operation in real-world conditions where ideal imaging is seldom achievable.

In real-world scenarios, images are often affected by multiple simultaneous degradations, e.g., noise and blur. To address this issue, unified models have been proposed that are capable of jointly handling multiple restoration tasks. These models offer two principal advantages. First, they are typically more computationally efficient than deploying separate models for individual tasks. Second, they reduce the risk of error propagation that may occur in sequential processing pipelines. For instance, in an image affected by both noise and blur, applying deblurring before denoising may amplify the noise or introduce spatial noise correlations, whereas performing denoising first may over-smooth the regions that are already blurred. Joint restoration models can overcome these limitations by learning to address multiple degradations holistically, thereby producing more accurate restoration.

2.2.1 Video restoration

A naive approach to video restoration involves applying an image restoration model independently to each frame. This strategy has two main drawbacks. First, it fails to leverage information from neighbouring frames, making the solution less effective. Depending on the type of degradation, adjacent frames can offer valuable support, e.g., providing additional noisy realisations of the same pixel (in denoising), capturing a sharper instance of the object (in deblurring), or showing the object at a larger scale (in super-resolution). Second, frame-by-frame processing tends to introduce noticeable variations between the restored frames, leading to visual artefacts such as flickering or jitter. These inconsistencies can make the video visually unpleasant. In practice, maintaining temporal consistency may be more important for the perceived video quality than maximising per-frame details.

Various techniques for handling motion in video restoration have been proposed. The most straightforward approach incorporates additional frames by concatenating them or their extracted features along the channel dimension, followed by standard convolutional processing. A special case of this strategy splits the model into two parts, i.e., first, each frame is restored individually and then temporal processing is applied to the restored results. While straightforward, this method suffers from a key limitation, i.e., independent, per-frame restoration can discard important information, thereby limiting the effectiveness of subsequent temporal processing. Moreover, as the number of concatenated frames increases, it becomes progressively harder for the network to exploit information from distant frames. To overcome this issue, cascaded approaches have been introduced. In such methods, the first stage processes each frame together with its immediate neighbours. In subsequent stages, the previously restored frames serve as input, allowing the information from

further temporal contexts to be gradually integrated. With each successive stage, the temporal receptive field expands.

The methods described above are constrained by their limited temporal receptive field. A natural extension to address this limitation is to incorporate recurrent mechanisms into the model. In this approach, the network processes not only the degraded frames but also the information from previous time steps, i.e., from its own earlier outputs. These outputs can be either the final restored frames or intermediate feature representations. When the current output depends on respective information from the preceding steps, the model is regarded as recurrent. In theory, recurrent architectures offer an unlimited temporal range. If the model relies solely on past information, it is referred to as unidirectional, making it suitable for online processing scenarios. Conversely, when both past and future frames are considered (in a bidirectional configuration), the model requires offline processing but can potentially achieve superior restoration performance. A potential drawback of recurrent neural networks (RNN) is error accumulation. Minor inaccuracies, typically negligible in single-pass processing, can be amplified over time through the recurrent mechanism, eventually generating severe and visually disturbing artefacts [154, 155].

Temporal Shift Modules [156], originally introduced for video understanding tasks, are also applied to video restoration [48, 46]. In each module, a portion of the channels is passed directly to the next layer, while other portions are additionally shifted either forward or backward along the temporal dimension. Despite operating on a limited number of frames, Temporal Shift Modules enhance the network's ability to capture longer-range temporal information as compared with standard 2D convolutions without increasing the computational cost for layers of the same shape. However, a notable drawback here is the introduction of latency, as each Temporal Shift Module contributes an additional frame delay.

To efficiently restore multiple frames, some authors propose processing them jointly [49, 157, 158, 50, 159]. This allows for the ongoing information exchange between the processed frames, similar to Temporal Shift Modules, but, depending on the implementation, it is not necessarily limited to adjacent frames. However, this approach introduces a significant delay, i.e., for the first output frame, the delay equals the number of input frames minus one. Additionally, the edge frames within each set are typically restored with lower quality, as they lack the temporal context of the closest and thus most useful neighbouring frames. To mitigate this issue, overlapping input sets are sometimes used, though this can reduce the frame rate and further increase the latency. Many state-of-the-art video restoration methods [160, 34] are designed to process entire sequences jointly, leveraging long-range temporal dependencies for the improved reconstruction quality. However, this design inherently restricts them to offline processing, as all the frames in the sequence must be available before the restoration can take place.

Handling motion between frames, particularly when it is significant, can be challenging for a network that lacks dedicated mechanisms. Even if the theoretical

receptive field of the network covers distant objects in the neighbouring frames, the effective receptive field may, in practice, be much smaller. The network may learn to focus on the reference frame during training, as this strategy more directly contributes to the performance improvement. Conversely, learning to handle complex motion can be substantially more difficult and, in the case of smaller networks, may not even be optimal. When the network capacity is limited, relying predominantly on the reference frame may yield better results.

Two mechanisms that do not explicitly estimate motion but can serve a similar purpose are dynamic kernels and inter-frame self-attention. In the case of dynamic kernels (also referred to as filter-adaptive or dynamic filtering), convolution weights are generated on-the-fly by a subnetwork based on the input features. This dynamic adaptation enables the models to adjust filter responses to the local content and motion characteristics. Dynamic kernels are successfully employed to such tasks as frame interpolation [161], video deblurring [162, 163], and video super-resolution [164]. They are also applied to burst denoising [165, 166] and burst super-resolution [167] which are both closely related to video processing yet generally exhibit less severe motion. Inter-frame self-attention captures dense dependencies between pixels across different frames. Using query embeddings from the reference frame and key embeddings from a supporting frame, attention weights are computed for each pixel pair. They are subsequently used to aggregate value embeddings from the supporting frame. This mechanism not only captures motion but can also discover multiple similar structures across the neighbouring frames. Self-attention, a powerful concept that revolutionised natural language processing and later computer vision, introduces quadratic complexity with respect to the input size, which makes it impractical for high-resolution videos. To address this, window-based approaches are proposed, where the inputs are partitioned into spatial windows, and attention is computed only within each window [168, 34].

The most straightforward way to handle motion between frames is by aligning them with optical flow. This enables even distant regions to be connected, which is particularly important for videos with large motion or high-resolution content, where motion extends across many pixels. The optical flow for alignment can be computed using either traditional algorithms or deep learning models. However, in both cases, the computations are resource intensive. Despite obtaining state-of-the-art results, deep models for optical flow face notable challenges. First, the distribution gap between the data used to train the optical flow model and the restoration input data can significantly degrade the flow quality, leading to inaccurate alignment. This gap may arise from the differences in motion types, colour characteristics, or brightness between training datasets and the actual use cases. Furthermore, degradations such as heavy noise or severe blur, which are often absent in optical flow training sets, can further reduce the alignment quality. Even when the flow is accurate, warping can introduce such artefacts as ghosting, oversmoothing when bilinear interpolation is applied or pixel repetition when nearest-neighbour sampling is used. Consequently, even perfect optical flow may be suboptimal for restoration tasks [169].

To address the above issues, several strategies have been proposed, including joint training of restoration and optical flow modules [169], creating specialised training datasets for flow estimation [170] and computing the flow on preliminarily restored frames. In some cases, rather than estimating dense optical flow, a global homography transformation between frames is applied [171, 172]. This approach is computationally cheaper, as it estimates a single geometric transform instead of dense pixel-wise motion representation, and it can help reduce certain artefacts such as ghosting.

An alternative explicit motion-handling approach is deformable convolution. Similar to dynamic kernels, it relaxes the constraints of standard convolutions. Instead of learning filter weights dynamically, deformable convolutions predict spatial offsets for sampling locations. In later variants [173], modulation mechanisms are added, allowing the model to adjust feature amplitudes. Although offsets often resemble optical flow, sometimes enforced through loss terms, deformable convolutions offer greater flexibility due to multiple learned kernel weights and modulation maps. Furthermore, using deformable convolutions is generally simpler than integrating optical flow. However, a significant drawback is their limited support, particularly on edge devices, which complicates the model deployment. A much simpler method to partially extend the receptive field for neighbouring frames is to apply constant spatial shifts, as proposed in ShiftNet [48]. However, since the shifts are predefined, the range of motion that can be effectively captured is limited, and computations associated with features shifted in directions not corresponding to the actual motion in the video are essentially wasted.

Inspired by classic methods, a combination of deep learning with non-local approaches is proposed. It relies on creating artificial supporting frames by assembling the patches from the neighbouring frames that are most similar to those in the reference frame [174, 175]. This strategy avoids explicit motion estimation by relying on a patch similarity search and benefits from the ability to exploit additional repetitions of similar patterns and textures across the data. A disadvantage is that finding similar patches across frames is a costly process, making non-local based methods inefficient.

2.2.2 Denoising

Denoising is the process of removing random, unwanted variations from a corrupted signal. In digital photography, noise originates from imperfections in the capture process and the particle nature of light. In computer-generated animation, it can result from a limited number of light rays used during rendering.

Sources and types of noise

The type of noise affects the choice of appropriate denoising solutions. Understanding its origin and characteristics is, therefore, essential.

In analogue photography, the main noise component is film grain, also called

film granularity. Traditional photography uses silver halide crystals, whose sensitivity to light increases with size. However, their non-uniform distribution causes brightness variations across an image. To enhance film sensitivity, larger crystals are used, but this results in more noticeable grain and reduced detail preservation.

In digital photography, silver crystals are replaced by the sensor's pixels arranged in a uniform grid. However, the fundamental trade-off remains: smaller pixels capture finer details but collect less light. Random fluctuations in photon arrival, the effect known as photon shot noise, become prominent in low-light conditions. Furthermore, increasing the ISO, which in digital cameras amplifies the electrical signal after capture, also leads to the noise increase, making high-ISO images particularly susceptible to visible noise. Since brightness information in digital cameras is transmitted as an electronic signal, fluctuations in voltage become an additional source of noise. In practice, various components of camera's signal-processing chain, such as readout, ISO amplification, and digitisation, contribute to the voltage fluctuations, causing deviations in RAW pixel values from their ideal values. This type of noise is commonly referred to as read noise.

Another source of noise arises from the thermal agitation of electrons within the photosensitive part of a pixel. This thermal activity can liberate electrons independently of photon capture, introducing thermal noise. While this noise is signal-independent, its impact increases with longer exposure times and higher temperatures.

A further artefact is the pixel response non-uniformity, where inherent variations in photon detection efficiency generates fixed-pattern noise. This pattern can be estimated by capturing the images under uniform illumination and subsequently utilized to calibrate and amend pixel sensitivity variations. Such correction is usually performed for digital single-lens reflex and metrology cameras.

Problems such as transmission errors, corrupted pixel elements in camera sensors or faulty memory locations can lead to salt-and-pepper noise, also called impulse noise. As an result, correct values are replaced with maximum (salt) or minimum (pepper) values, which in grayscale images generates white and black dots.

Finally, some difference between the perfect RAW value and the obtained one is introduced by digitisation of an analogue voltage signal. This quantisation error, defined as the difference between the signal value and the rounded one, is sometimes referred to as quantisation noise. Its impact is usually negligible, especially for sensors with precision higher than eight bits.

Because collecting clean–noisy image pairs for training deep learning models is challenging, it is common to use synthetic noise models during training. Shot noise, also known as Poisson noise, is typically modeled using the Poisson distribution. A key insight is that the noise level increases with the signal but not proportionally, it grows more slowly than the signal. As a result, the noise becomes more noticeable when strong amplification is applied, typically in low-light images. Within a given image, it tends to be more visible in brighter regions than in darker ones. Sensor read

noise and thermal noise are typically modeled using a zero-mean normal (Gaussian) distribution. To model salt-and-pepper noise, it is common to assume a certain probability of each pixel retaining its original value. For the remaining cases, the corrupted pixel is equally likely to be set to either the maximum or the minimum possible value.

Datasets

Image denoising methods are commonly benchmarked using images with synthetically added noise. Early approaches often relied on small datasets such as Kodak24 [176] or BSD68 [32], which are sufficient for evaluating non-trainable algorithms. However, with the rise in deep learning, larger datasets have become essential for training, while smaller datasets are retained for validation and benchmarking. Popular datasets containing high-resolution images include Flickr2K [177], DIV2K [178], and Large Scale Dataset for Image Restoration [179].

Because synthetic noise does not always accurately reflect real-world disturbances, several datasets with clean-noisy image pairs have been introduced, i.a., Darmstadt Noise Dataset [180], the Smartphone Image Denoising Dataset [181], the Nam dataset[182], the PolyU dataset [183], and the Renoir dataset[184]. Many of them provide RAW images, as denoising is often the most effective when applied prior to other image processing steps, although some datasets contain RGB images. These datasets differ in terms of the camera models used, the number of scenes, ISO levels, the exposure times, and the procedures used to compute the ground truth.

Typically, the ground truth is estimated by averaging multiple (up to 1000) captures of the same scene. However, in very dark or very bright regions, this averaging may be inaccurate due to clipping which distorts the noise distribution. To address this issue, the authors of the Smartphone Image Denoising Dataset propose the Robust Mean Image Estimation technique. Other methods involve capturing reference images with lower ISO and correspondingly longer exposure times. Some authors also compensate for the motion between captures, remove outliers with brightness inconsistencies or excessive movement, and normalise the intensity across the scene. Since noise is especially prevalent in low-light conditions, it is also important to mention datasets developed for low-light enhancement such as the LOw-Light dataset [185], the See-in-the-Dark dataset [186], and the Single Image Contrast Enhancement dataset [187].

Classic Approaches

Before the advent of deep learning, image denoising was addressed using a variety of techniques. A large family of these methods adopts a Bayesian perspective, where the objective is to estimate the most probable denoised image, given a noisy observation. One of the most common formulations is Maximum A Posteriori estimation, which incorporates the prior term describing the likelihood of a clean image independently of a noisy observation. Designing effective prior terms is a major focus in denoising methods grounded in the Maximum A Posteriori formulation. [130, 188, 189].

Because noise primarily affects high-frequency image components, many meth-

ods operate in transform domains, including wavelet thresholding [190, 191], discrete cosine transform filtering [192, 193], and approaches based on curvelet [194, 195] or contourlet [196, 197] transforms. Another major category consists of spatial filtering methods. A simple approach is to average the pixel values in a local window, which reduces the noise, but also smooths out image details. Gaussian filtering [198] improves this approach by weighting the pixels based on their spatial distance. Bilateral filtering [199] further enhances edge preservation by also considering the colour similarity when computing weights. Median filtering [200, 201, 202] proves to be effective in removing salt-and-pepper noise.

These spatial filtering techniques evolved into non-local methods which exploit the self-similarity of image patches across the entire image. Notable examples include Non-Local Means [203] and BM3D [204], which group similar patches and perform collaborative filtering.

Deep learning methods

Following the success of deep learning in high-level vision tasks, such as image classification and semantic segmentation, researchers began exploring its potential for low-level vision tasks, including image denoising. Although early work demonstrated the feasibility of using neural networks for denoising [205, 206, 207], a significant breakthrough was achieved with the introduction of DnCNN [32]. This model integrates modern architectural components such as batch normalisation and global residual learning. The latter, now widely adopted in image restoration, involves predicting the noise component, i.e., the residual between the noisy and the clean image, rather than the clean image itself. The estimated noise is then subtracted from the noisy input to obtain the denoised output. The next notable milestone is FFDNet [208]. It introduces the concept of a noise level map which provides an additional input channel encoding the noise characteristics of the image. Models utilising such information are referred to as unblind models, in contrast to blind ones, which operate without any explicit knowledge of the noise level. In scenarios where noise characteristics are unavailable, models such as CBDNet [209] incorporate a dedicated subnetwork to estimate the noise level. In addition to purely data-driven methods, several approaches explore the integration of classical image processing techniques with deep learning frameworks. These include the incorporation of non-local similarity search [210], learned kernel regression for adaptive filtering [211], and Fourier transform-based representations [212]. Nevertheless, progress in image denoising is principally driven by the adoption of generic deep learning advancements: deeper architectures, novel modules, e.g., attention mechanisms, improved normalisation and activation functions, and transition toward more powerful architectures such as Transformers and Mamba. Recently, top-performing denoising models in denoising represents the family of general-purpose image restoration networks, such as Restormer [213] and SwinIR [214], designed to handle a range of degradation types beyond just noise.

An important research direction concerns training image denoising models

in such scenarios where acquiring clean, ground-truth images is infeasible. The Noise2Noise [36] approach demonstrates that, under certain statistical assumptions, a network can be trained to perform nearly as well as when using clean targets by replacing the clean image with another noisy realisation of the same scene. The Noise2Noise approach is extended to the scenarios when capturing multiple realisations of the same scene is not possible, e.g., Noise2Self [37], Noise2Void [39], Self2Self [215], Neighbor2Neighbor [216].

2.2.3 Video Denoising

The demand for video denoising methods is increasing alongside the growing volume of video content being created and transmitted. In applications such as teleconferencing and telemedicine, it is essential to develop efficient denoising solutions that minimise frame delay and enable real-time data transmission. In contrast to single-frame denoising, video denoising can achieve superior results by exploiting temporal redundancy between frames. Importantly, this enhancement in quality can often be realised without substantial increases in computational overhead or processing delay.

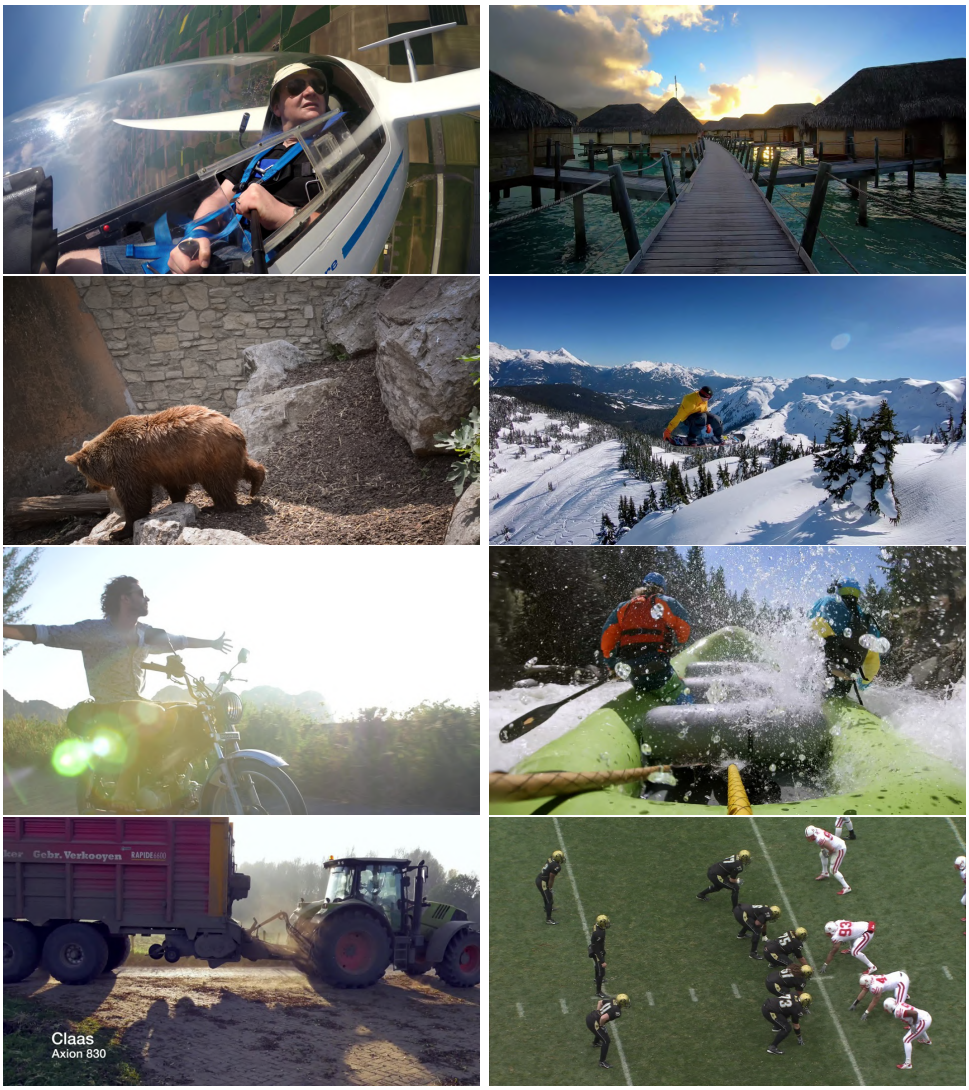
Datasets

Similar to image denoising, video denoising methods are often benchmarked using clean video sequences corrupted with artificially added noise. A widely adopted benchmark is the DAVIS dataset [217] originally developed for the video object segmentation task. To train video denoising models, the set is commonly divided into a training set and a validation set, comprising 30 and 86 sequences respectively. Most sequences have a resolution of 854×480 pixels. DAVIS includes a variety of scenes and motion patterns, including both camera movements and dynamic objects, and features well-illuminated content. In addition to DAVIS, the Set8 dataset [218] is frequently used for testing purposes. It is a collection of eight video sequences, most with a resolution of 960×540 pixels. It includes both sequences with relatively static scenes and examples exhibiting significant motion. For evaluation, noise with the same standard deviations is added. Example frames from both datasets are presented in Figure 2.1.

A noisy signal is typically simulated using additive Gaussian noise with standard deviation values of up to 50 (on an intensity scale ranging from 0 to 255):

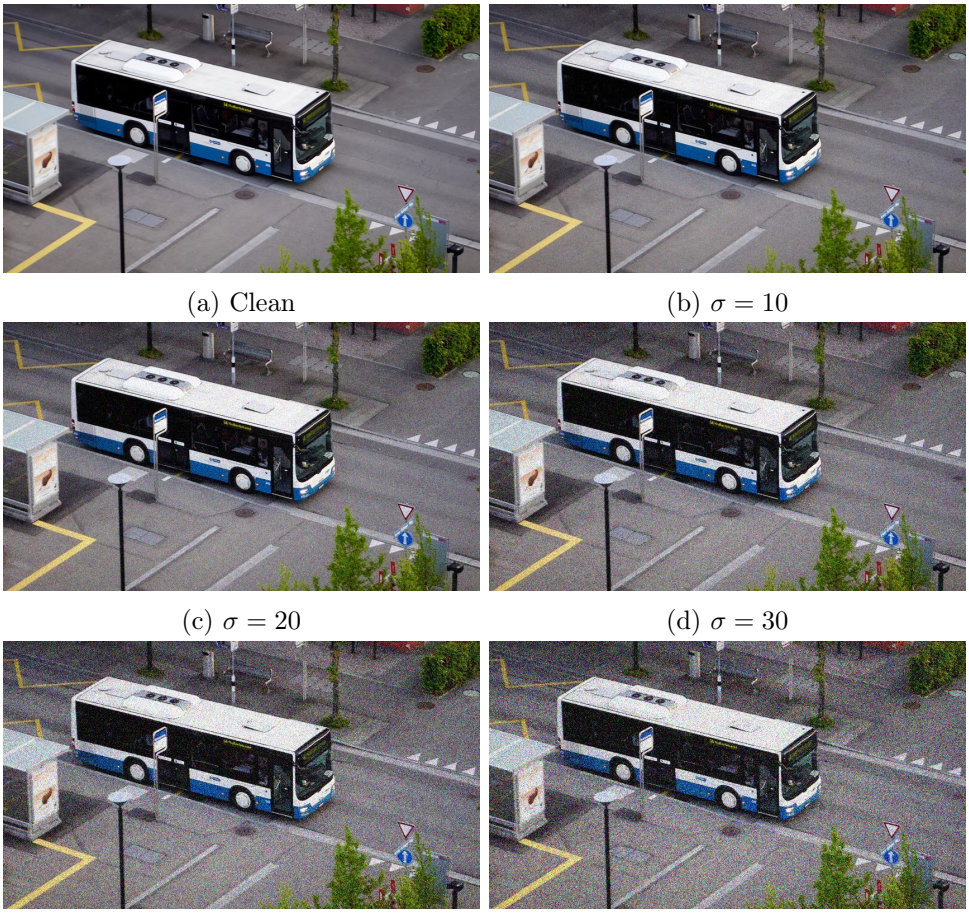
$$x = y + \mathcal{N}(0, \sigma^2) \tag{2.1}$$

where x and y denote the corrupted and clean signals respectively, $\mathcal{N}$ denotes the Gaussian distribution, and $\sigma \in (0, +\infty)$ is the standard deviation of the Gaussian noise component. An exemplary frame corrupted with noise sampled using different σ values is presented in Figure 2.2.



(a) DAVIS (b) Set8

Figure 2.1: Examples from DAVIS (left column) and Set8 (right column) datasets.



(a) Clean (b) $\sigma = 10$

(c) $\sigma = 20$ (d) $\sigma = 30$

(e) $\sigma = 40$ (f) $\sigma = 50$

Figure 2.2: Frame corrupted with Gaussian noise sampled with different σ values.

For more realistic noise simulation, the Poisson-Gaussian distribution can be employed

$$x = \Pi_{[0,U]}(\alpha\mathcal{P}((y - b_g)/\alpha) + \mathcal{N}(0, \sigma^2) + b_g) \tag{2.2}$$

where $\mathcal{P}$ denotes the Poisson distribution, $\alpha \in (0, +\infty)$ is a scaling parameter, and $b_g \in (0, +\infty)$ represents a background value. $\Pi_{[L,U]}$ denotes the projection onto the interval, defined as $\Pi_{[L,U]}(x) = min(max(x, L,), U)$, where L is the minimum representable value and U is the maximum representable value.

Capturing real noise with the corresponding clean counterparts is particularly challenging for video data. In dynamic scenes, it is nearly impossible to record multiple well-aligned noisy frames required to compute a ground truth through averaging, due to the object or camera motion. Capturing a clean frame with longer exposure times also leads to scene changes and often introduces motion blur, making it unsuitable as ground truth.

To address this problem, several strategies have been explored. The first involves capturing scenes with minimal inherent motion, where dynamic elements are manually repositioned between views to simulate motion, while remaining static during individual captures to ensure accurate alignment and averaging. This approach is exemplified by the CRVD [143]. CRVD includes recordings of 11 indoor scenes, mainly with a static camera and a single moving object. The videos are stored in RAW format, preserving the original Bayer pattern. For each scene, recordings are made at five ISO levels (1600, 3200, 6400, 12800, 25600). At each ISO, seven different viewpoints are captured, and for each viewpoint, multiple (from 150 to 500, depending on the ISO level) noisy frames are recorded. The ground truth frames are obtained by averaging noisy frames and, to suppress residual noise, additional denoising is applied using BM3D [204]. In the released version of the dataset, 10 noisy frames and the clean one are available for each viewpoint. CRVD is further extended with longer noisy outdoor sequences intended for the visual evaluation of the trained models. This extension consists of 50 sequences (10 sequences per ISO level), each containing 50 frames. For each ISO level, the authors provide variance and scaling factor parameters for the corresponding Poisson–Gaussian noise model. Additionally, to pre-train their model, the authors synthesised RAW data from the MOT dataset [219] using the same noise model.

To construct the See-Moving-Objects-in-the-Dark dataset [220], designed for low-light enhancement with paired noisy low-light and clean bright frames, a dual-camera setup combined with a beam splitter was employed. One of the cameras was further equipped with a neutral density filter to reduce brightness for the low-light capture. Although effective, this approach has several drawbacks. It is difficult to implement using high-end Digital Single-Lens Reflex cameras; it is susceptible to misalignment arising from imperfections in the positioning of the cameras and the beam splitter; and, particularly in the context of denoising, bright frames may contain overexposed regions, resulting in saturated pixels that do not provide reliable ground-truth information.

The Seeing Dynamic Scene in the Dark dataset [221] adopted a different strategy by employing an electric slide rail to facilitate repeatable camera motion. In a single pass, low-light noisy sequences were recorded, followed by their clean and bright counterparts in a subsequent pass. This approach also introduces slight misalignments between the paired sequences and is limited to simple horizontal camera motion with static objects. Similar to See-Moving-Objects-in-the-Dark dataset, the method suffers from difficulties in handling overexposed areas, which reduces its suitability for denoising.

Another widely used approach involves recapturing the displayed sequences. This allows for recording scenes with complex motions while enabling playback to be paused at any moment to capture multiple frames for averaging. This strategy was first introduced in the Low-light Realistic Motion Video Dataset [222] and later adopted in the Recaptured Raw Video Denoising Dataset (ReCRVD) [223] thus offering greater flexibility in terms of the recorded motion diversity and camera setups as compared with the previously described methods.

The ReCRVD, a dataset specifically designed for video denoising, comprises 120 sequences, each recorded at one of five ISO levels. Most sequences originate from the DAVIS dataset, and the authors propose a split of 90 sequences for training and 30 for validation, with an even distribution of ISO levels across the sets. The dataset is recorded using the same image sensor as CRVD and maintains the same five ISO levels, making it a natural extension of the earlier dataset. Contrary to CRVD, the clean reference images in ReCRVD are generated by capturing frames with lower ISO and long exposure time. To compensate for spatial misalignment and brightness difference between clean and noisy frames, tailored post-processing is applied. Visual comparison between CRVD and ReCRVD is presented in Figure 2.3. The first two columns present two consecutive noisy frames from various sequences. The third column shows a zoomed-in patch from the second frame, while the fourth column displays its clean counterpart. In CRVD, the objects exhibit rapid motion between frames, whereas the background remains static. In contrast, ReCRVD features more natural motion, with both background and foreground moving smoothly. The green tint is characteristic for RAW images.

Classic approaches

Before the emergence of deep learning methods, the most widely adopted approaches for video denoising relied on exploiting non-local spatio-temporal similarity. Among these, V-BM4D [224] is one of the most popular and well-known representatives of this class. The algorithm operates in several stages. First, spatio-temporal volumes are constructed by combining a reference patch from the noisy frame with patches from the neighbouring frames, selected with a motion estimation procedure. In the subsequent grouping stage, similar volumes are gathered through a non-local search. For each group, a separable linear transform is applied, the transformed coefficients are attenuated, i.e., thresholded or Wiener filtered, and the inverse transform is then performed to reconstruct the denoised volumes. Because the grouped volumes

Figure 2.3: Examples from CRVD (top three rows) and ReCRVD (bottom three rows) datasets.

overlap, the final aggregation step combines the multiple estimates, leveraging this redundancy to reduce residual noise and artefacts. In addition to non-local approaches, extensions of other image denoising techniques to the video domain are proposed, including filtering-based [225, 226] and frequency-domain methods [227, 228].

Deep learning methods

Early attempt to apply deep learning to video denoising employed RNNs to model temporal dependencies [229]. While this approach demonstrates the feasibility of temporal modelling, it is constrained to grayscale video and fails to match the performance of the established classical methods. A significant advancement came with ViDeNN [230], which introduces a two-stage framework consisting of a spatial denoising module and a temporal refinement one. In the first stage, the spatial denoiser is trained on a large dataset of noisy-clean image pairs. The temporal module is then trained separately to exploit temporal redundancy by refining the obtained outputs using neighbouring, preliminarily denoised, frames. However, because the system is not trained in end-to-end fashion, the pipeline cannot fully compensate for potentially lost image details in the initial denoising stage, which may limit its reconstruction fidelity. Similar to ViDeNN, DVDNet [218] consists of two modules, namely a spatial and a temporal denoising network, and the model is trained in two stages. Unlike ViDeNN, however, neighbouring frames are aligned using optical flow, specifically DeepFlow [231]. The authors place greater emphasis on the spatial denoising module, designing it with approximately twice the number of layers compared to the temporal one.

An important milestone in the use of motion estimation within video denoising was introduced by TOFlow [169]. This model is the first to incorporate a deep-learning-based optical flow network directly into the video denoising pipeline. The optical flow component is first pre-trained on the Sintel dataset [232], then fine-tuned on noisy video data, and ultimately the entire network is trained in the end-to-end manner. Crucially, the authors observe that the flow fields optimised for restoration differ substantially from traditional optical flow, and that learning motion representations tailored to the denoising task yield superior results. In contrast to TOFlow, FastDVDnet [45] does not rely on explicit motion estimation. The model utilises a two-stage cascaded architecture. In the first stage, three identical U-Net blocks independently process overlapping triplets of consecutive frames, namely frames $t-2$ to t, $t-1$ to $t+1$, and t to $t+2$, to generate preliminary denoised estimates of the central frame. These three intermediate outputs are then passed on to a second-stage U-Net block, which has an identical architecture but separate parameters, to fuse the information and produce the final denoised output. All the U-Net modules incorporate global residual learning, where the network predicts the noise component which is subsequently subtracted from the central frame. This approach demonstrates a favourable trade-off between accuracy and computational efficiency, establishing FastDVDnet as an important baseline for subsequent video denoising research. Another method that also forgoes explicit motion estimation for the sake of efficiency is EMVD [47], in which a simple yet effective recurrent scheme fuses the

current noisy frame with a pre-denoised version of the previous one. The blending parameters, i.e., weight maps applied to both frames before the summation, are estimated with lightweight convolutional module. This method efficiently suppresses noise in static regions but struggles in the areas with significant motion. Additionally, the authors introduce the use of two invertible transforms to decorrelate colour and frequency information from the input frames, further enhancing the denoising performance. Other methods, e.g., MMNet [49], ReMoNet [158], and ASwin [233], employ simultaneous processing of multiple frames, which proves to be effective in balancing denoising performance with computational efficiency, although the introduced latency remains a drawback. ReMoNet integrates this approach with a recurrent mechanism, while ASwin leverages an efficient attention mechanism. TempFormer [52] implements an intermediate solution in which a subsequence of video frames is denoised together, based on a longer subsequence centered at the same frame. Together with the proposed striding, which increases the number of the shared input frames and leads to overlapping outputs, such a design enables the use of an *Overlap Loss*. This loss enforces similarity between the corresponding outputs from various temporal blocks, resulting in improved temporal consistency. In BSVD [46] and ShiftNet [48], at multiple processing stages, the computed features are split and respective parts are assigned to specific tasks. The majority of these features are used to obtain further features for the respective frames while smaller portions are employed in the restoration of the neighbouring frame. In ShiftNet, these features are additionally spatially shifted in multiple directions, which expands the effective receptive field and allows for the improved implicit temporal alignment. In parallel, deeper models incorporating explicit motion estimation have also been explored. EDVR [234], originally developed for super-resolution and deblurring, has been successfully adapted for video denoising. It introduces the Pyramid, Cascading and Deformable alignment module, which performs multi-scale feature alignment via deformable convolutions in a coarse-to-fine manner. Notably, this alignment operates without explicit supervision and still yields effective frame alignment. A similar module is used in RViDeNet [143], the first network designed to perform RAW video denoising. In addition to the alignment module, the authors propose the use of channel, spatial and temporal attention modules. Temporal fusion is performed independently for each colour channel derived from RAW data packing. The derived features are subsequently combined in a spatial fusion module. Before RViDeNet, there was only one attempt related to RAW video denoising [235]. Its authors propose a single-image network which is executed on two captures of the same scene during training. To enforce temporal consistency, they use an additional loss term that imposes similarity on the outputs for both images. BasicVSR++ [160], originally designed for super-resolution, adopts a hybrid strategy for handling motion. It computes optical flow between frames to guide the offsets used in deformable convolution. Specifically, neighbouring frames are initially warped using optical flow to compensate for large motion. Residual offsets are then estimated based on the warped neighbouring features and the current frame's features, and the final offsets are computed by summing these residuals with the original flow. For the denoising

variant, the authors improve the computational efficiency by performing most operations at lower resolutions, including downsizing the frames by a factor of up to four before computing the optical flow [51]. A key feature of the model is its bidirectional feature propagation, where information is propagated both forward and backward across the video sequence. It is designed to process entire video sequences collectively, making the model particularly well-suited for offline video restoration, where all the frames are accessible during inference. Video Restoration Transformer [34] further advances this line of work by combining flow-guided deformable alignment with transformer-based architectures. It performs feature alignment within pairs of neighbouring frames using a self-attention mechanism. In Recurrent Video Restoration Transformer [35], the authors propose Guided Deformable Attention to align clips, which are groups of neighbouring frames processed jointly, with each other. The model processes the clips in a unidirectional recurrent fashion, which reduces latency compared to the earlier methods that require global bidirectional processing of entire sequences. Another approach that combines recurrent mechanisms with partial access to future information is FloRNN [236]. In this method, a look-ahead module leverages the nearest future frames by aligning their recurrently computed features to the frame currently being processed using optical flow. This design allows the network to exploit temporal information from both past and near-future contexts while maintaining lower latency than fully offline bidirectional methods.

2.3 Literature Gaps

While frame alignment gets a lot of focus in the field of video restoration, techniques proposed in state-of-the-art methods are not adjusted to the applications requiring lightweight processing. For the models without explicit temporal alignment, there is usually no mechanism to encourage the utilisation of neighbouring frames, leading to bias toward focusing on the reference frame. To mitigate this problem, a new training paradigm is proposed in this dissertation. It forces the model to focus on neigbouring frames by withholding information from the central frame during the initial pre-training stage. The behavior learned at this stage is subsequently developed when full information is made available.

When large motions occur, handling them effectively generally requires dedicated alignment mechanisms, particularly in the case of lightweight models. The existing explicit motion estimation techniques are not typically tailored to such models, with the notable exception of BasicVSR++ in the context of denoising and deblurring. However, in case of BasicVSR++, the impact of downsizing frames prior to optical flow estimation is not analysed in isolation, and the range of downsizing ratios considered is limited up to a factor of four. In this study, the effect of varying the input resolution to the optical flow network on video denoising performance is investigated. It is demonstrated that even coarse alignment obtained from the optical flow computed on heavily downsized frames (by up to a factor of 12) can still provide substantial quality improvements and is applicable to lightweight models .

Chapter 3

Frame Interpolation for Pre-Training Video Denoising Models

3.1 Introduction

In this chapter, the first component of the first hypothesis is addressed. Video denoising models often tend to rely heavily on the reference frame while underutilising information from the neighbouring frames. Leveraging temporal information is challenging in realistic scenarios due to the presence of varying motion between frames which requires a significant portion of the model's capacity to be dedicated to the motion compensation. As a result, during training, models may follow an easier but suboptimal path, focusing primarily on the central frame. To encourage greater utilisation of neighbouring frames, a pre-training strategy is proposed in which the video denoising model is first trained on a related task where the central frame is not available, namely frame interpolation. Furthermore, to reduce the task gap between frame interpolation and video denoising, an intermediate training stage is introduced in which the information from the central frame is gradually reintroduced.

3.2 Method

To restore a noisy frame from time step t, the noise (residual) n_t is usually estimated and subtracted from the corrupted frame x_t. Noise n_t can be expressed as:

$$n_t = x_t - y_t \tag{3.1}$$

where y_t is a clean frame at time step t. Deep video denoising of the noisy frame x_t using this frame with the neighbouring ones $X_t = \{x_{t-k}, \ldots, x_t, \ldots, x_{t+j}\}$ (where $k, j \geq 0$ denote the number of past and future frames) and the noise map m, is

usually performed using the residual function

$$\tilde{y}_t = x_t - N(X_t, m) \tag{3.2}$$

where $\tilde{y}_t$ denotes the estimated, restored frame, and N is a neural network. This approach can be extended to tree-like architectures that involve multiple blocks $\mathcal{B}$ containing a single network N and the residual connection (3.2) for the central frame. Such an architecture of D networks arranged in a tree-like structure is defined as $\mathcal{M}_\Theta = \{\mathcal{B}_\tau^d \mid d = 1, \ldots, D\}$. Notably, the blocks with the center input frames at times $\tau(d) = \{t - d + 1, \ldots, t, \ldots, t + d - 1\}$ are indexed as $\mathcal{B}_\tau^d$ at depth d. All the blocks from a single depth level d share the same weights and are thus indexed as N^d. To extract early features from each input independently, each network begins with group convolutions. The noise map m is shared across the blocks, and it is therefore dropped from further equations for simplicity. Blocks $\mathcal{B}_\tau^d$ process three consecutive input frames into a single output

$$y_\tau^d = \mathcal{B}_\tau^d(i_{\tau-1}^d, i_\tau^d, i_{\tau+1}^d) = i_\tau^d - N^d(i_{\tau-1}^d, i_\tau^d, i_{\tau+1}^d) \tag{3.3}$$

where $i_\tau^D = x_\tau$, while intermediate inputs for the blocks at depths $d < D$ are $i_\tau^d = y_\tau^{d+1}$, and the final denoising result $\tilde{y}_t$ estimated at depth $d = 1$ is given by y_t^1. The described architecture is trained by minimising loss

$$\mathcal{L}_{\text{den}} = \psi\left(\tilde{y}_t, y_t\right) \tag{3.4}$$

where ψ denotes a quality measure, e.g., ℓ_1 or ℓ_2. For the proposed architecture, the number of employed past and future frames is equal, and $k = D$. Architectures with depths $D = 1$ and $D = 2$ are visualised in Figure 3.1.

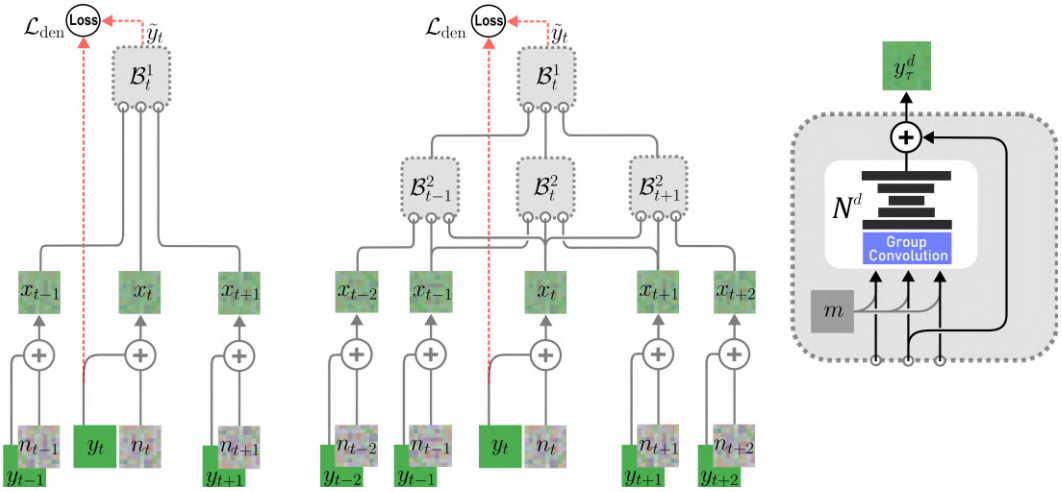


Figure 3.1: Architectures employed for FIP validation with $D = 1$ (left), $D = 2$ (middle), and block $\mathcal{B}^d$ (right).

Such an architecture, when considered in isolation has computational complexity $O(D^2)$ but, following [237], a substantial portion of the blocks outputs in the

architectures with depth greater than one can be reused in subsequent time steps. As a result only D blocks need to process new inputs, which reduces the inference complexity to $O(D)$.

Overview

For the architecture described above, it is proposed to pre-train the model on the frame interpolation task. Out of E training epochs during the initial E_{FIP} epochs, the model is trained to interpolate the masked central frame of the input sequence. This encourages the model to rely primarily on past and future frames. Afterwards, during the Post-FIP epoch, the model is gradually transitioned to the standard denoising task and finally, for the remaining epochs, it is trained under standard settings.

Progressive noising Interpolating a frame is even more challenging when the inputs are noisy. To facilitate this, the noise in the unmasked frames is progressively increased. Throughout the pre-training epochs e ($1 \leq e \leq E_{\text{FIP}}$), the noise is scaled by value growing linearly from 0 to 1:

$$\hat{x}_\tau = y_\tau + \mathcal{G}_1(n_\tau, e) \tag{3.5}$$

according to the first gate function

$$\mathcal{G}_1(n_\tau, e) = n_\tau \cdot (e-1)/(E_{\text{FIP}} - 1). \tag{3.6}$$

Input swapping When $D > 1$, multiple blocks occur at depth D. Since they share weights, it is necessary to mask the central input frame also for other bottom blocks $\mathcal{B}^D_\tau$ instantiated at $\tau \neq t$. To ensure that the central, interpolated frame is not passed on to them, the inputs are swapped according to the following rule:

$$y^D_\tau = \begin{cases} \mathcal{B}^D_\tau(\hat{x}_{\tau-1},\ \mathbf{0}\ , \hat{x}_\tau\ \ ) & \text{if } \tau < t \\ \mathcal{B}^D_\tau(\hat{x}_{\tau-1},\ \mathbf{0}\ , \hat{x}_{\tau+1}) & \text{if } \tau = t \\ \mathcal{B}^D_\tau(\hat{x}_\tau\ \ ,\ \mathbf{0}\ , \hat{x}_{\tau+1}) & \text{if } \tau > t \end{cases} \tag{3.7}$$

where $\mathbf{0}$ denotes a tensor of the same shape as the input frame, but filled with zeros.

Loss terms To facilitate the interpolation task, the loss function during the pre-training stage is augmented with additional terms that compare the outputs of the network blocks at time step t with the corresponding clean image

$$\mathcal{L}_{\text{int}} = \psi(\tilde{y}_t, y_t) + \sum_{d=2}^{D} \psi(y^d_t, y_t). \tag{3.8}$$

For simplicity, the same measure ψ is used. FIP for the architecture with $D = 2$ is visualised in Figure 3.2.

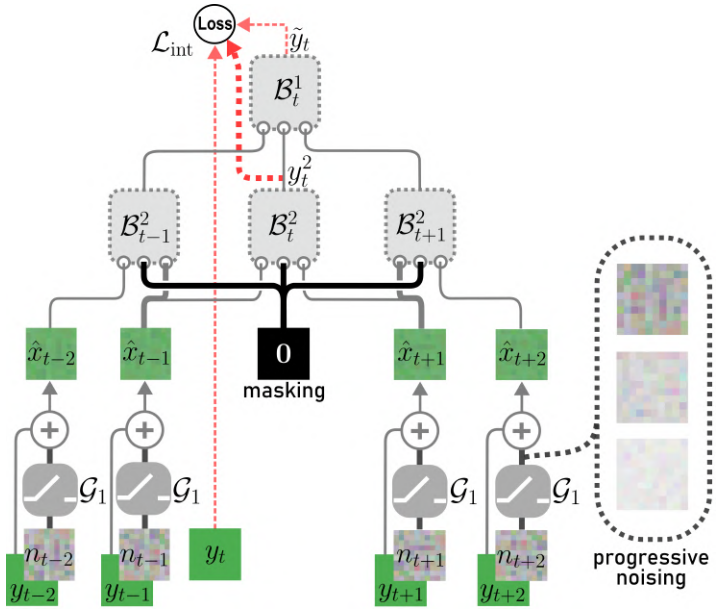


Figure 3.2: Visualisation of FIP for the architecture with $D = 2$.

Post-FIP To facilitate the transition from the pre-training phase (FIP) to the standard denoising training, a Post-FIP phase consisting of a single epoch is introduced. During this epoch, the network is trained on the denoising objective (3.4), and the bottom blocks are fed with unmasked frames, following the standard (un-swapped) configuration. In the FIP phase, the residual connection in the bottom blocks $\mathcal{B}_\tau^D$ passes zero values (i.e., $\mathbf{0}$). To ensure a smoother transition, in the Post-FIP phase, the frame passed through the residual connection is gradually reintroduced by scaling it with a factor that grows linearly from 0 to 1 across the epoch. The modified bottom blocks $\overline{\mathcal{B}}_\tau^D$ are defined as

$$y_\tau^D = \overline{\mathcal{B}}_\tau^D(x_{\tau-1}, x_\tau, x_{\tau+1}, it) = \mathcal{G}_2(x_\tau, it) + N^D(x_{\tau-1}, x_\tau, x_{\tau+1}, i) \tag{3.9}$$

with the value of the second gate function

$$\mathcal{G}_2(x, it) = x \cdot it/I \tag{3.10}$$

increasing linearly over iterations $0 \leq it < I$ in the Post-FIP epoch comprising I iterations. The Post-FIP stage is visualised in Figure 3.3.

3.3 Experiments

Datasets The presented experiments are based on four datasets. For synthetic Gaussian denoising, two standard RGB datasets are included: DAVIS [217] and Set8 [218]. Set8 contains sequences with higher spatial resolution and larger motion magnitudes than those in the DAVIS dataset. This is demonstrated by the histograms of motion magnitudes computed based on optical flow which are presented in Figure

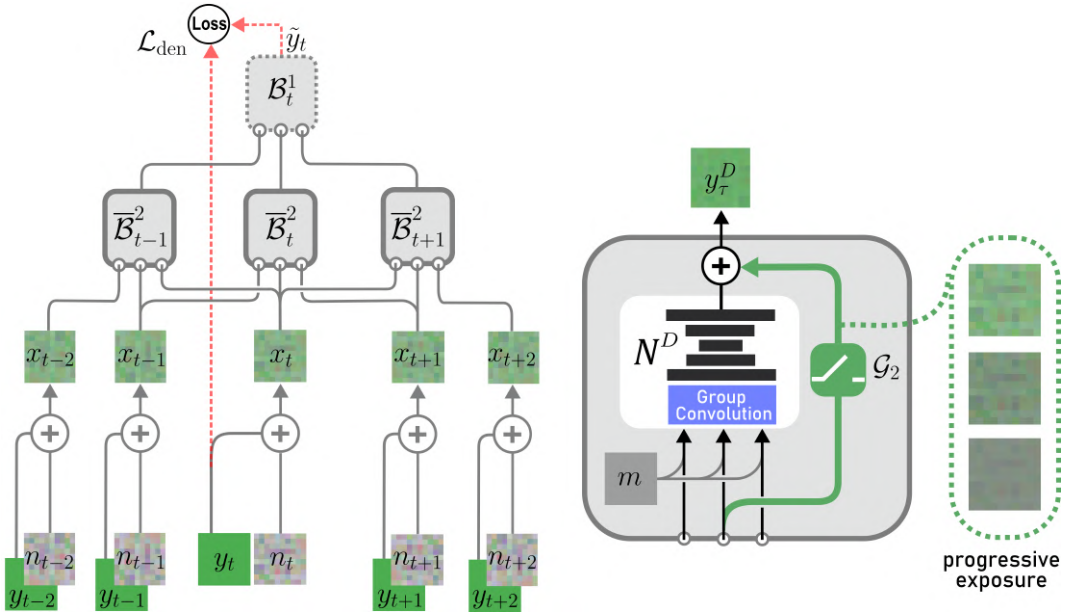


Figure 3.3: Visualisation of the Post-FIP stage for the architecture with $D = 2$ (left) and block $\overline{\mathcal{B}}^D$ (right).

3.4. DAVIS sequences are divided into the training and test sets following the standard split, while Set8 is used only for the testing purposes. For both datasets, the results are presented on five levels of Gaussian noise ($\sigma = 10, 20, 30, 40, 50$). Additionally, the models are trained and evaluated on RAW datasets with real noise, i.e., CRVD [143] and ReCRVD [223].

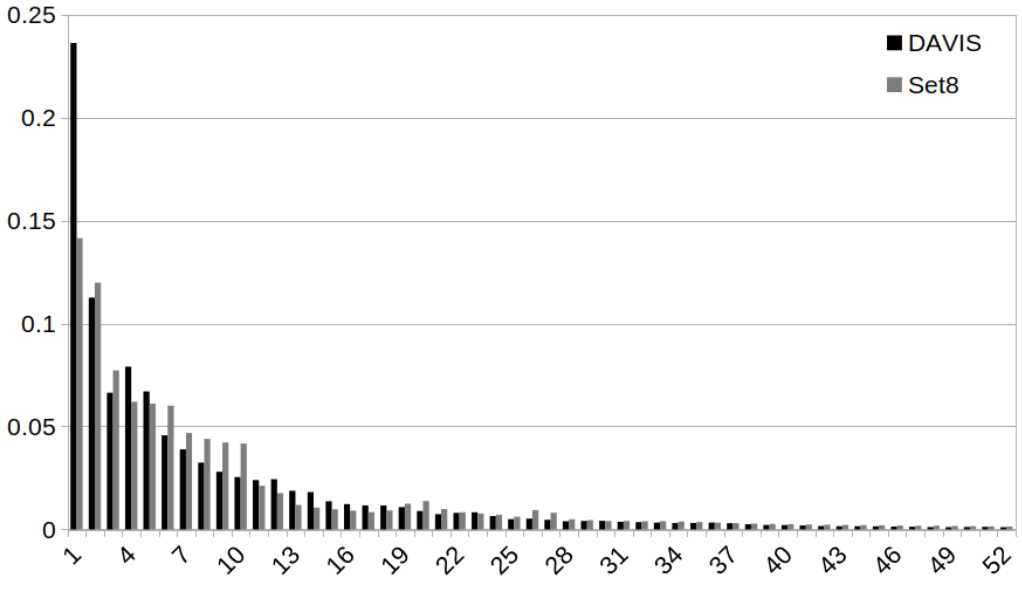


Figure 3.4: Motion histograms for the DAVIS and Set8 datasets.

Evaluation metrics To evaluate the per-frame quality of video denoising, three standard metrics are used, i.e., Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) [238], and Learned Perceptual Image Patch Simi-

larity (LPIPS) [239], while Warping Error (E(W)) [131] is used to assess its temporal consistency. To compare the models in terms of their efficiency, frames per second (FPS) computed on an NVIDIA GPU Quadro RTX 5000, giga multiply–accumulate operations (GMACs), and frame latency are used. To further analyse the models behavior in terms of temporal context usage, two additional measures are proposed: Single Frame Difference (SFD) and Percentage of Correct Intrinsic Flow (PCIF). SFD measures the performance drop caused by replacing the sequence elements with the central frame:

$$\begin{aligned} \mathrm{SFD} = \mathrm{PSNR}(\mathcal{M}_\Theta(x_{t-k}, \ldots, x_t, \ldots, x_{t+k}, m), y_t) - \\ \mathrm{PSNR}(\mathcal{M}_\Theta(x_t \quad, \ldots, x_t, \ldots, x_t \quad, m), y_t). \end{aligned} \tag{3.11}$$

It is assumed that models relying more heavily on the information from neighbouring frames are more sensitive to the loss of additional frames. Consequently, such models are expected to exhibit higher SFD scores. The main advantages of SFD measure are its simplicity and the ability to compute it over a large number of samples within a reasonable time.

To gain deeper insights, a second measure is introduced. To exploit temporal information, models must find corresponding points in the neighbouring frames. This task is particularly challenging for the models without explicit motion estimation. However, it has been demonstrated that deep learning models can leverage temporal information in such scenarios [45, 240]. To assess which points most strongly influence the denoising result, gradient-based techniques can be incorporated. Following the previous attempts to use gradients for visualisation purposes [240], the first-order Taylor decomposition of the analysed temporal model is employed, i.e.,

$$\tilde{y}_{t,l} \approx \langle \nabla \mathcal{M}_\Theta(X_t, m), X_t \rangle + b \tag{3.12}$$

where b denotes a scalar bias and l represents the pixel location. The gradients $\nabla \mathcal{M}_\Theta$ are computed with respect to X_t using backpropagation from $\tilde{y}_{t,l}$. After averaging over the colour channels, they produce gradient maps $(g_{t-k}, \ldots, g_{t+k})$. Based on the gradient map, a motion vector referred to as intrinsic flow can be computed as the weighted average of positions $p \in \mathbb{R}^2$ with the largest gradient values, shifted by l

$$u_{t\pm k,l} = \frac{\sum_{p \in P} |g_{t\pm k,p}| \cdot (p - l)}{\sum_{p \in P} |g_{t\pm k,p}|} \tag{3.13}$$

where $P = \{p \mid g_{t\pm k,p} > 0.2 \cdot \max(g_{t\pm k})\}$ (refer to [240]).

For the models without explicit motion estimation, achieving precise implicit temporal alignment is essential to enhance video denoising. To quantify this precision, the obtained intrinsic flow vectors can be compared with ground truth displacements $v_{t\pm k,l}$ obtained from optical flow. It is assumed that if the difference between them is below a reasonable low threshold r_1, then the denoised pixel can effectively benefit from temporal information. To exclude trivial motions or emphasise more

challenging ones, the points with the optical flow magnitude below a lower threshold r_2 are disregarded. Additionally, to constrain the measurement to a specific motion range, an upper threshold r_3 can be applied. Finally PCIF is computed as

$$\text{PCIF} = \frac{\#\{\|u_{t\pm k,l} - v_{t\pm k,l}\| < r_1\}}{\#Loc} \times 100, \quad \forall\, l \in Loc \tag{3.14}$$

where $Loc = \{l \mid r_2 < \|v_{t\pm k,l}\| \leq r_3\}$ for $0 < r_1 \leq r_2 < r_3$ and $\#$ denotes the size of the set.

Training details The proposed architecture $\mathcal{M}_\Theta$ is implemented using five different networks N: Unet [45], NAFnet [241], ADFNet [242], Restormer [213] and CGNet [243]. At the beginning of each network, a group convolution layer fed with three frames is inserted. Since the research focus is on efficient solutions, the model sizes are reduced (NAFnet from default 17M to 3.9M, Restormer from 26M to 1.4M, CGNet from 167M to 22M). The original network implementations are used and all the models are trained for $E = 80$ epochs. The resulting models are referred to using the name of the backbone network and the architecture depth, e.g., Unet-D2. For the models trained with FIP, the same total number of epochs is maintained, with the initial $E_{\text{FIP}} = 10$ epochs dedicated to pre-training followed by a single Post-FIP epoch. For the Unet-based models, the original training settings, including learning rates, batch size, and data preprocessing, from [45] are retained. For the remaining denoisers orthogonalisation is not applied, while the losses, optimisers, and learning rate schedules are adopted from the respective network training configuration. For Gaussian denoising, the noise map is created using the standard deviation σ of synthesized noise. For real noise, the standard deviation σ and scaling parameter α estimated by the authors of CRVD [143] are used for noise map creation.

3.3.1 Ablations

To evaluate the effectiveness of the proposed techniques, the Unet-D2 architecture corresponding to FastDVDnet [45] is employed.

In Figure 3.5, the qualitative interpolation results of the Unet-D2 during the FIP phase are presented. They demonstrate that the proposed network is capable of performing interpolation to a meaningful extent, which indicates its ability to estimate and compensate for motion. The output of the bottom block y_t^2 already captures the overall structure of the interpolated frame, while the final result $\tilde{y}_t$ provides sharper details and improved visual fidelity. Moreover, the interpolation quality consistently improves throughout the pre-training epochs, despite the increasing noise level in the input frames.

The ablation study of the proposed training scheme is presented in Table 3.1. The reported values are averaged over three training runs and five Gaussian noise levels. The best results are highlighted in **bold black**, the second-best results - in **bold gray**. The upper part of the table presents the results of applying the

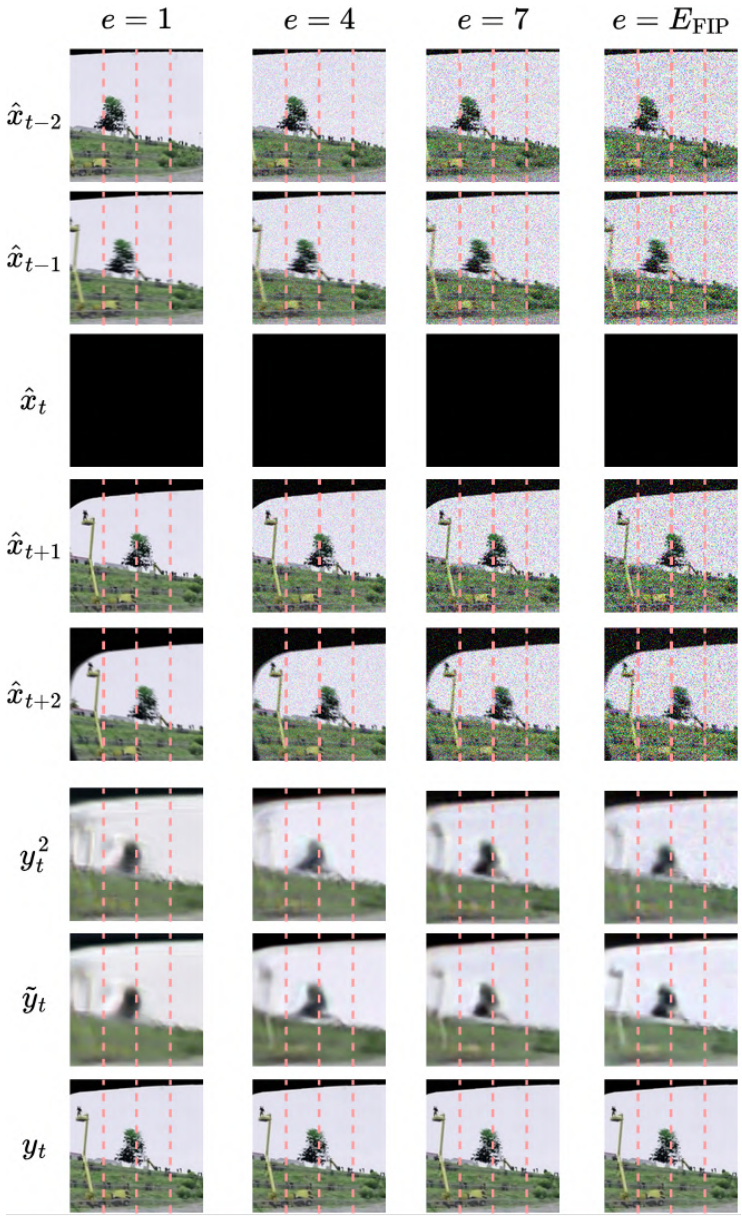


Figure 3.5: Intermediate qualitative results obtained during the FIP phase over multiple epochs on the DAVIS dataset.

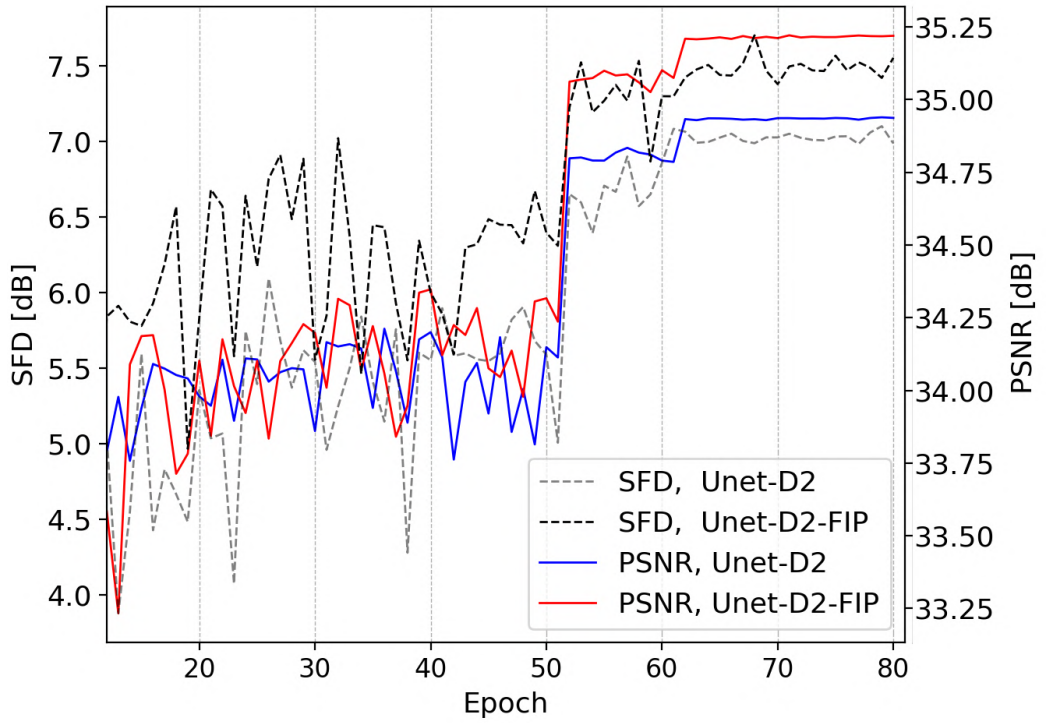
Table 3.1: Ablation study results for Unet-D2 on the DAVIS dataset.

Abl. no.	FIP	Interm. L. E	Interm. L. E_{FIP}	Group Conv	$\mathcal{G}_1$	$\mathcal{G}_2$	E(W)↓	PSNR↑	SSIM↑	LPIPS↓	SFD↑
1	✗	✗	✗	✓	✗	✗	0.0162	34.87	0.9202	0.1046	7.14
2	✗	✓	✗	✓	✗	✗	0.0165	34.72	0.9177	0.1093	5.85
3	✗	✗	✓	✓	✗	✗	0.0159	34.95	0.9215	0.1034	5.54
4	✓	✗	✗	✓	✓	in res.	0.0158	35.06	0.9234	**0.0992**	7.72
5	✓	✗	✓	✗	✓	in res.	0.0160	34.95	0.9214	0.1021	7.42
6	✓	✗	✓	✓	✗	in res.	0.0159	34.98	0.9222	0.1005	7.49
7	✓	✗	✓	✓	✓	✗	0.0163	34.86	0.9197	0.1041	7.03
8	✓	✗	✓	✓	✓	after N	0.0157	35.04	0.9232	0.1037	7.68
9	✓	✗	✓	✓	✓	before res.	**0.0156**	**35.08**	**0.9237**	**0.1004**	**7.84**
10	✓	✗	✓	✓	✓	in res.	**0.0155**	**35.13**	**0.9244**	0.1010	**7.80**

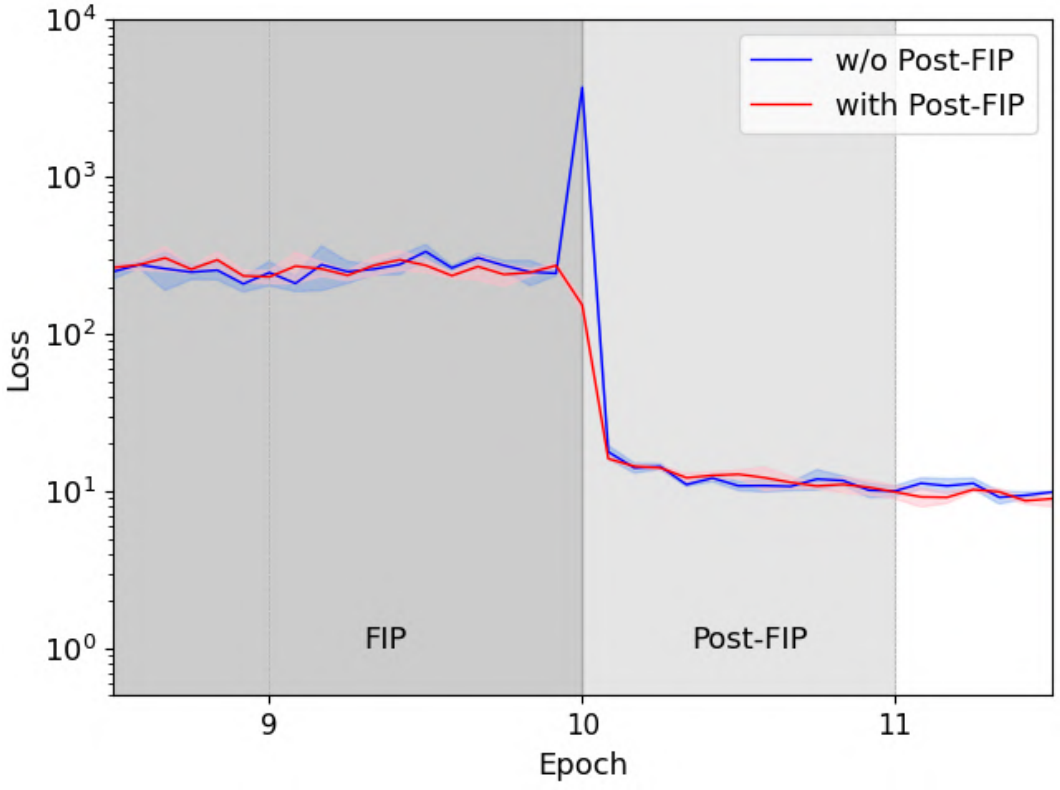
intermediate loss term in training without the interpolation stage. Incorporating this loss throughout all the epochs (E) negatively affects all the metrics, whereas restricting it to the initial E_{FIP} epochs yields improvements in both per-frame quality and temporal consistency. The lower part of the table compares the models utilising the proposed pre-training approach. The target configuration (abl. 10) achieves the best results in most metrics. Removing the intermediate loss from this version (abl. 4) leads to the degradation across all the metrics except LPIPS. Replacing group convolutions with standard convolutions (abl. 5) or removing progressive noising (abl. 6) reduces the performance to the level of the model trained without FIP (abl. 3). Finally, the impact of the gating function G_2 is analysed. The model without it (abl. 7) performs on par with the baseline Unet-D2, representing a drop compared to the Unet-D2 trained with the intermediate loss. Applying G_2 to the output of N (abl. 8) or to the central input of the bottom blocks before the residual connection (abl. 9) yields improvements over the standard training, but smaller than those achieved with the proposed method.

Figure 3.6 presents the training curves for selected ablation variants. In Figure 3.6a, the baseline model Unet-D2 is compared with the final version trained with FIP. Validation PSNR and SFD curves across the epochs are shown. It can be clearly observed that, from the very beginning of the denoising stage, the SFD score is significantly higher for the FIP version, thus indicating the improved utilisation of temporal information. This advantage is maintained throughout the training, and the enhanced exploitation of the temporal context also translates into a higher denoising quality. In Figure 3.6b, the impact of including the Post-FIP epoch is illustrated. Compared to a direct transition to the denoising stage, the training loss decreases more smoothly, without a sharp initial peak. This results in more stable training dynamics, avoiding abrupt parameter updates that could otherwise lead to forgetting the temporal-context utilisation developed during the pre-training phase.

Figures 3.7 and 3.8 present gradient-based visualisations illustrating the effect of FIP. In Figure 3.7, the computed gradient maps (shown as heatmaps overlaid on grayscale past frames) and estimations of a single intrinsic flow vector $u_{t-1,l}$ (black



(a) Validation PSNR and SFD with and without FIP.



(b) Training loss with and without Post-FIP epoch.

Figure 3.6: Training curves for Unet-D2 variants on DAVIS dataset.

arrow) between the central and past frame for the reference point l ($\times$) are presented. The ground-truth motion is represented by the optical flow vector (white arrow), obtained with RAFT [244]. This example demonstrates that the model trained with FIP handles motion more effectively. In Figure 3.8, gradient maps, averaged over all the positions in the central frame, are shown. For improved visualisation, the maps are normalised either with respect to the frames at temporal delay k (upper left) or with respect to the central frame (lower right). The model trained with FIP is observed to exploit the neighbouring frames, particularly x_{t-2} and x_{t+2}, to a greater extent and to make a better use of the near-edge regions in contrast to the baseline.

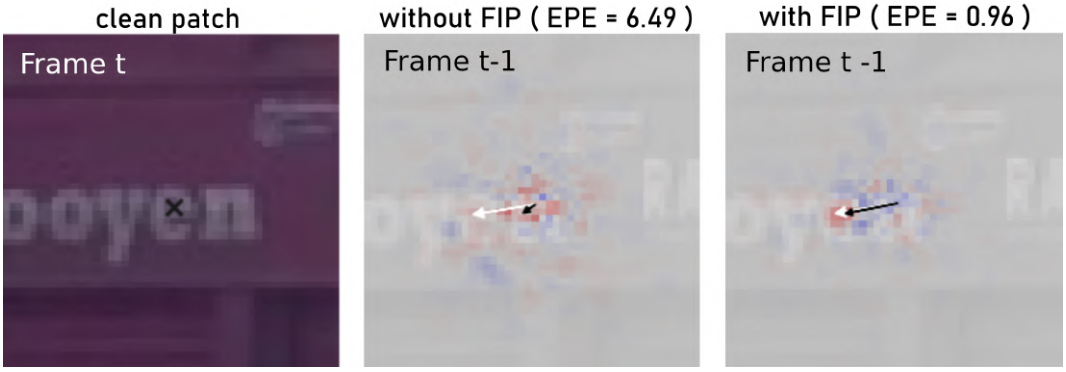


Figure 3.7: Comparison of intrinsic flow vectors between models trained with and without FIP.

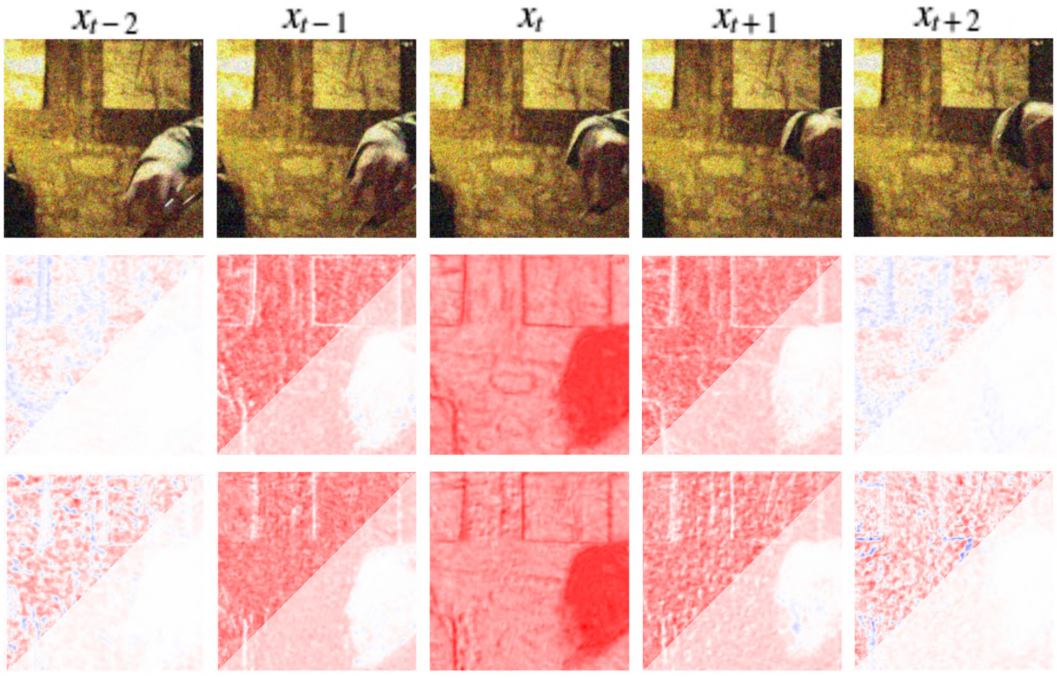


Figure 3.8: Comparison of averaged gradient maps between models trained without (middle row) and with (bottom row) FIP.

Table 3.2 presents the results evaluating the impact of FIP duration (number of epochs E_{FIP}). Among the tested values, $E_{\text{FIP}} = 10$ achieves the best perfor-

Table 3.2: Impact of FIP duration on denoising performance on the DAVIS dataset.

E_{FIP}	E(W)↓	PSNR↑	SSIM↑	LPIPS↓
5	0.0160	34.97	0.9218	0.1031
10	**0.0155**	**35.13**	**0.9244**	0.1010
15	0.0158	35.01	0.9225	0.1024
20	0.0160	34.97	0.9221	**0.1007**

Table 3.3: Comparison of Temporal Augmentation and FIP on the DAVIS dataset.

FIP	TA	E(W)↓	PSNR↑	SSIM↑	LPIPS↓
✗	✗	0.0162	34.87	0.9202	0.1046
✗	✓	0.0163	34.93	0.9212	**0.1010**
✓	✗	**0.0155**	**35.13**	**0.9244**	**0.1010**
✓	✓	0.0156	35.05	0.9232	0.1045

mance across most metrics. Table 3.3 presents the results of a further comparison of the proposed approach with another method designed to improve temporal information utilisation, i.e., Temporal Augmentation [136]. Temporal Augmentation is implemented by subsampling every second frame in 25% of the training sequences, thereby enriching the dataset with examples containing larger motion. While Temporal Augmentation improves the performance compared to the baseline training, it remains less effective than FIP alone. Combining Temporal Augmentation with FIP achieves better results than Temporal Augmentation by itself, but does not surpass the effectiveness of FIP. It is hypothesised that the presence of larger motions may make the interpolation task excessively difficult for certain sequences, which in turn hinders the development of the desired behaviour during the FIP phase.

3.3.2 Results

In Table 3.4, the effect of using FIP is presented across multiple architectures (five different networks N and two values of D) on two datasets, i.e., DAVIS and Set8. For most architectures, the version with FIP outperforms the baseline in all the metrics. Only in a few cases, the baseline achieves slightly better LPIPS values. Across all the networks N, the impact of FIP is more pronounced for $D = 2$ than for a shallower version. These observations are further supported by the qualitative results presented in Figure 3.9. In Table 3.5, PSNR values for individual noise levels are presented for D2 architectures for all the examined networks. Using FIP improves the results across noise levels, but it tends to have a greater impact for stronger noise levels. It is established [240] and intuitive that models rely more on more distant pixels where the noise is stronger. Better implicit motion estimation helps to utilise more pixels in a more precise way. Next, the impact of FIP for the selected models (Unet-D2 and CGNet-D2) with respect to the motion magnitude is evaluated (Table 3.6). To this end, frames are divided into spatial regions where pixels move within five predefined ranges (mov1–5). The ranges $0-1$, $1-3$, $3-6$, $6-14$, and > 14 pixels are chosen so that each interval contains approximately the same number of pixels

in the DAVIS validation set. Motion magnitudes are estimated using the optical flow computed with the RAFT model [244]. Across all the ranges, FIP consistently improves the accuracy. However, the relative gain diminishes for the fastest motions, particularly for Unet-D2, whose receptive field is relatively limited.

Table 3.4: Impact of FIP on denoising performance across models and datasets.

Model		DAVIS				Set8			
		E(W)	PSNR	SSIM	LPIPS	E(W)	PSNR	SSIM	LPIPS
Unet [45]	D1	0.0178	34.42	0.912	**0.108**	0.0290	32.43	0.888	**0.106**
	D1-FIP	**0.0174**	**34.47**	**0.913**	0.110	**0.0286**	**32.48**	**0.889**	0.108
	D2	0.0162	34.87	0.920	0.105	0.0274	32.77	0.896	0.103
	D2-FIP	**0.0155**	**35.13**	**0.924**	**0.101**	**0.0265**	**32.93**	**0.899**	**0.102**
NAFnet [241]	D1	0.0171	34.51	0.914	0.115	0.0283	32.52	0.890	**0.110**
	D1-FIP	**0.0163**	**34.62**	**0.916**	**0.113**	**0.0273**	**32.63**	**0.892**	**0.110**
	D2	0.0158	34.86	0.919	0.110	0.0270	32.77	0.895	**0.107**
	D2-FIP	**0.0149**	**35.10**	**0.924**	**0.105**	**0.0259**	**32.97**	**0.899**	0.108
Restormer [213]	D1	**0.0160**	34.81	0.918	0.110	**0.0270**	32.82	**0.896**	0.108
	D1-FIP	**0.0160**	**34.85**	**0.919**	**0.108**	**0.0270**	**32.85**	**0.896**	**0.105**
	D2	0.0150	35.22	0.925	0.100	0.0258	33.14	0.903	0.101
	D2-FIP	**0.0142**	**35.50**	**0.930**	**0.093**	**0.0254**	**33.26**	**0.908**	**0.101**
ADFNet [245]	D1	0.0168	34.73	0.917	0.103	0.0280	32.71	0.894	0.100
	D1-FIP	**0.0155**	**34.96**	**0.923**	**0.096**	**0.0266**	**32.85**	**0.897**	**0.098**
	D2	0.0153	35.18	0.925	0.096	0.0283	32.62	0.892	0.105
	D2-FIP	**0.0142**	**35.45**	**0.930**	**0.090**	**0.0255**	**33.16**	**0.904**	**0.093**
CGNet [243]	D1	0.0152	35.17	**0.925**	**0.098**	0.0263	33.01	**0.900**	0.102
	D1-FIP	**0.0151**	**35.19**	**0.925**	0.099	**0.0262**	**33.06**	**0.900**	**0.099**
	D2	0.0139	35.63	0.932	0.092	0.0253	33.33	0.906	0.094
	D2-FIP	**0.0137**	**35.79**	**0.934**	**0.087**	**0.0250**	**33.39**	**0.907**	**0.093**

To validate the impact of FIP on implicit temporal alignment, a further gradient-based analysis is performed for the architectures with various networks and $D = 2$. In Figures. 3.10 and 3.11, dense visualisations of, respectively first- and second-order, intrinsic flows are presented. For comparison, the corresponding optical flows are also presented. The first- and second-order flows correspond to the maps computed between the central frame and the frames $t - 1$ and $t - 2$ respectively. It can be observed that the intrinsic flow computed for the larger model, i.e., CGNet-D2, is closer to the ground-truth optical flow than its Unet-D2 counterpart. Nevertheless, in both cases, the use of FIP improves the intrinsic flow estimation, especially for the second-order case. Additional results for other networks further confirm the positive impact of pre-training on implicit motion estimation.

To evaluate intrinsic flows quantitatively, 200 pixel locations l are selected from

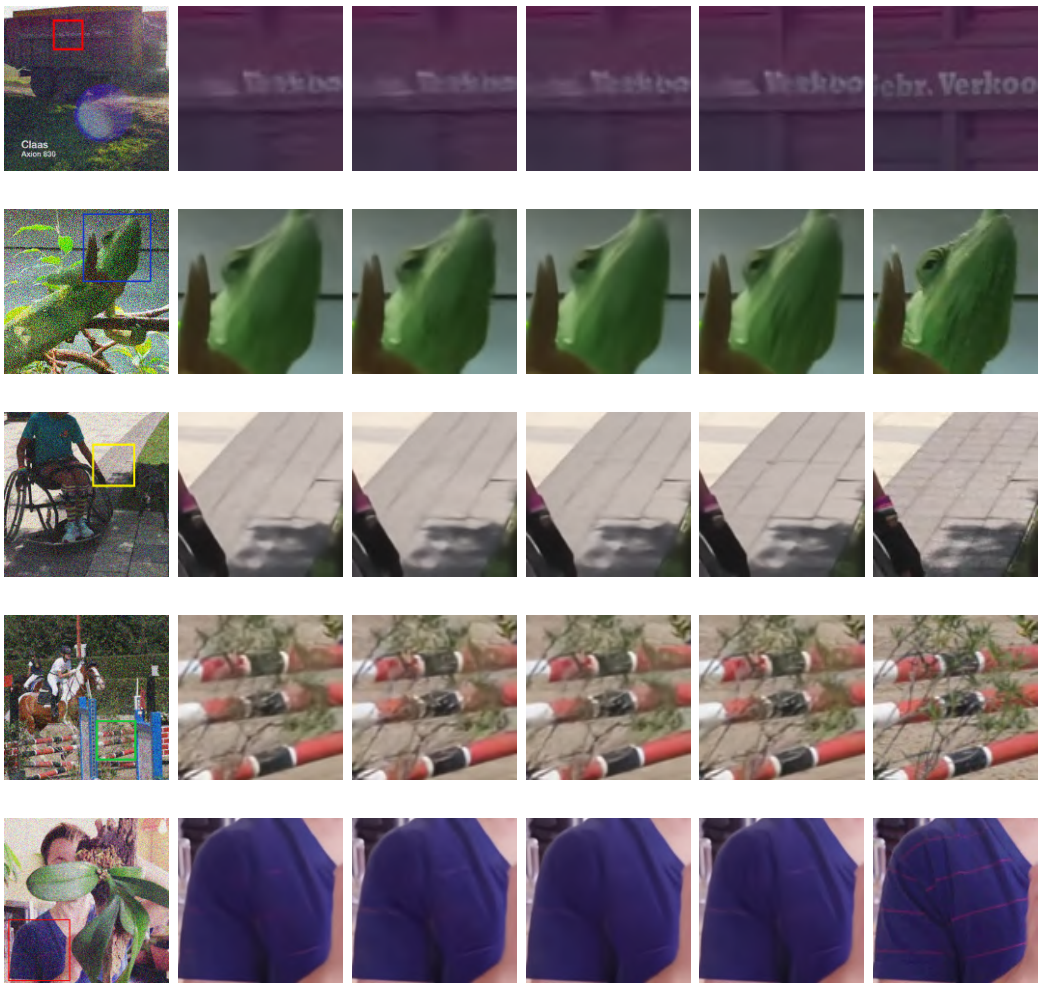


Figure 3.9: Qualitative results for models employing various networks N: (from top to bottom) Unet, NAFnet, Restormer, ADFnet and CGNet.

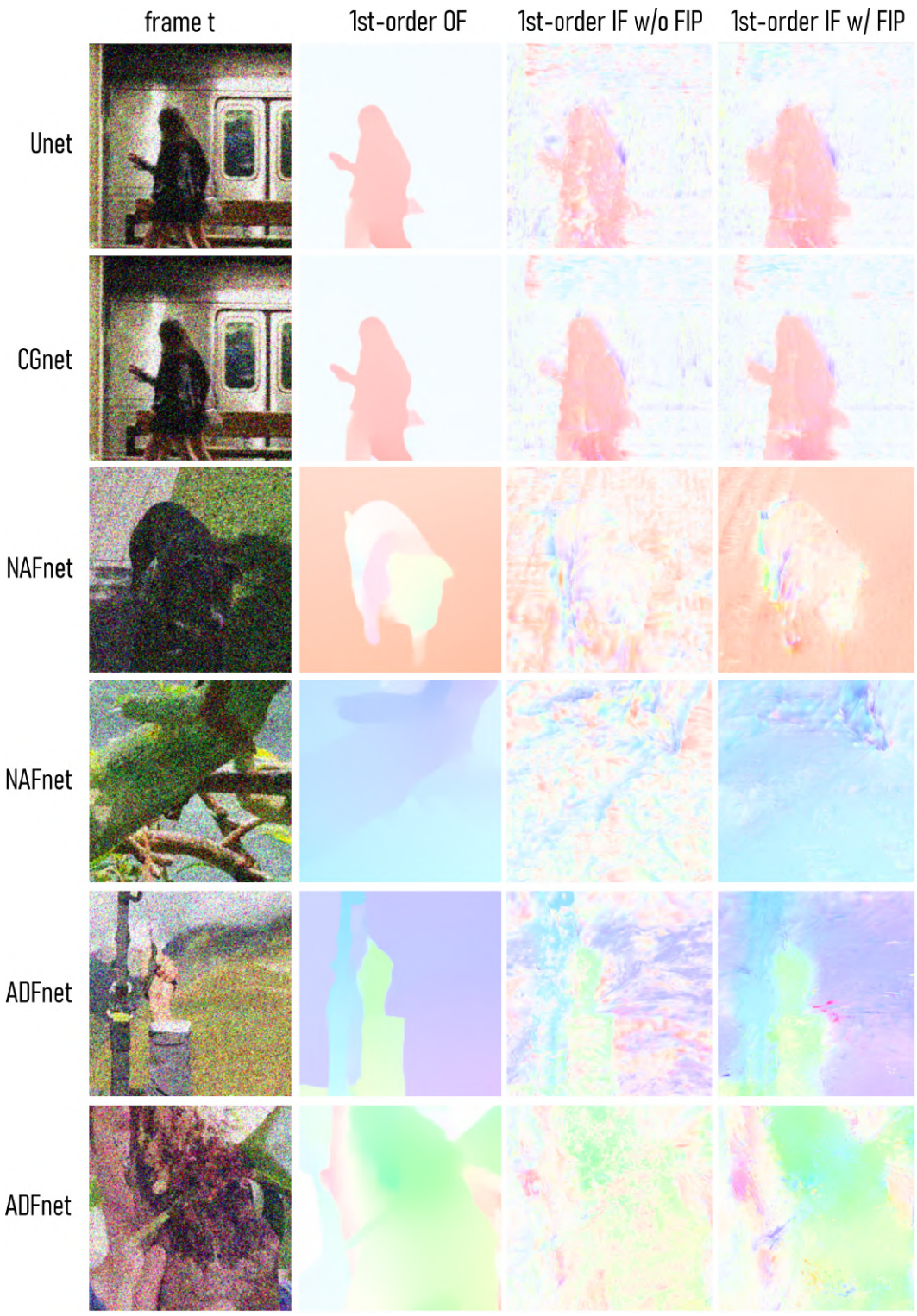


Figure 3.10: Comparison of first-order intrinsic flows across models.

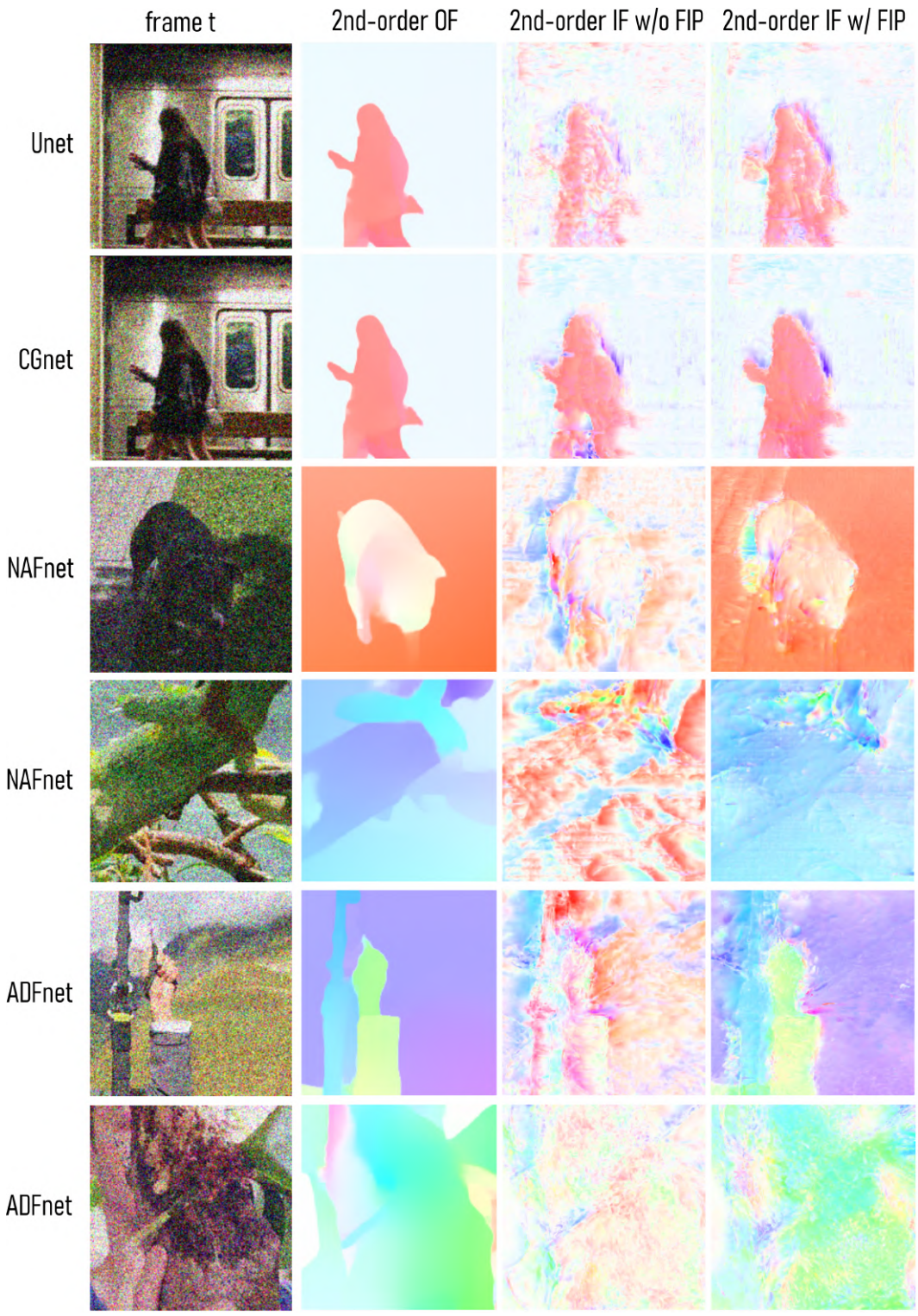


Figure 3.11: Comparison of second-order intrinsic flows across models.

Table 3.5: Impact of FIP on denoising performance across noise levels.

	DAVIS, σ =					Set8, σ =				
Model	10	20	30	40	50	10	20	30	40	50
Unet D2	39.23	36.03	34.21	32.93	31.94	37.38	33.95	32.05	30.74	29.75
Unet D2-FIP	39.44	36.30	34.48	33.20	32.21	37.48	34.10	32.21	30.91	29.93
Δ	0.21	0.27	0.27	0.27	0.27	0.10	0.15	0.16	0.17	0.18
NAFnet D2	39.28	36.00	34.16	32.90	31.96	37.48	33.93	32.00	30.70	29.73
NAFnet D2-FIP	39.41	36.21	34.43	33.20	32.26	37.58	34.12	32.23	30.94	29.99
Δ	0.13	0.22	0.26	0.29	0.31	0.10	0.19	0.23	0.25	0.25
Restormer D2	39.56	36.38	34.56	33.29	32.32	37.72	34.32	32.42	31.12	30.14
Restormer D2-FIP	39.71	36.63	34.86	33.62	32.66	37.77	34.41	32.55	31.27	30.30
Δ	0.15	0.25	0.30	0.33	0.34	0.04	0.09	0.13	0.15	0.16
ADFNet D2	39.46	36.31	34.53	33.29	32.33	37.56	34.14	32.26	30.97	30.00
ADFNet D2-FIP	39.65	36.60	34.82	33.57	32.60	37.65	34.31	32.45	31.17	30.21
Δ	0.19	0.28	0.29	0.28	0.27	0.09	0.17	0.19	0.20	0.20
CGNet D2	39.82	36.76	35.00	33.76	32.80	37.85	34.48	32.61	31.33	30.37
CGNet D2-FIP	39.97	36.93	35.16	33.92	32.96	37.90	34.54	32.67	31.40	30.44
Δ	0.16	0.16	0.17	0.17	0.17	0.05	0.06	0.06	0.07	0.07

Table 3.6: Impact of FIP on denoising performance across motion ranges.

	DAVIS					Set8				
Model	mov1	mov2	mov3	mov4	mov5	mov1	mov2	mov3	mov4	mov5
Unet-D2	36.13	34.79	33.92	33.60	34.02	34.90	33.44	33.12	32.43	33.50
Unet-D2-FIP	**36.38**	**35.16**	**34.30**	**33.85**	**34.04**	**35.03**	**33.63**	**33.34**	**32.60**	**33.53**
CGNet-D2	36.77	35.53	34.69	34.35	34.34	35.32	33.91	33.73	32.99	33.67
CGNet-D2-FIP	**36.85**	**35.69**	**34.87**	**34.54**	**34.47**	**35.39**	**34.00**	**33.80**	**33.07**	**33.71**

each DAVIS test sequence, constrained to lie within a predefined optical flow range (2–20). The first-order intrinsic flows are assessed using PCIF with a threshold of $r_1 = 2$ over multiple 1-pixel motion ranges, as illustrated in Figure 3.12. Again, it can be observed that the intrinsic flow estimated by CGNet-D2 outperforms that of Unet-D2. In particular, for the motions with magnitudes greater than 14, Unet-D2 fails to capture the motion reliably, while CGNet-D2 successfully handles the majority of such large displacements. More importantly, across all the motion ranges, FIP consistently yields higher-quality intrinsic flows for both networks. For example, Unet-D2 trained with FIP correctly estimates approximately 45% of the motions in the 7–8 pixel range compared to less than 5% for the baseline model. Furthermore, the evolution of PCIF throughout training epochs is analysed (Figure 3.13, top). For this and all the subsequent validations, PCIF is computed without division into motion ranges with $r_2 = 2$ and $r_3 = 20$. Unet-D2 with FIP learns to estimate motion earlier. Already during the initial E_{FIP} epochs, its PCIF surpasses that of the baseline model. Additionally, immediately after the Post-FIP epoch, model

exhibits a pronounced increase, achieving an approximately three times higher PCIF level than the baseline model. By the end of the training, the PCIF of the model trained with FIP nearly doubles, reaching almost 50% compared with only about 25% for the baseline.

Next, PCIF under different noise levels is examined (Figure 3.13, bottom left), where stable values across the conditions are observed. Finally, PCIF across varying intrinsic flow orders (from one to three) is compared between CGNet-based architectures with different depths D (Figure 3.13, bottom right). Again, the models trained with FIP consistently achieve higher values in all the configurations. Moreover, FIP enables a consistent improvement in PCIF with increasing depth, which is not always the case for the baseline models. In particular, without FIP, the PCIF of the second-order intrinsic flow in CGNet-D3 is lower than that in CGNet-D2.

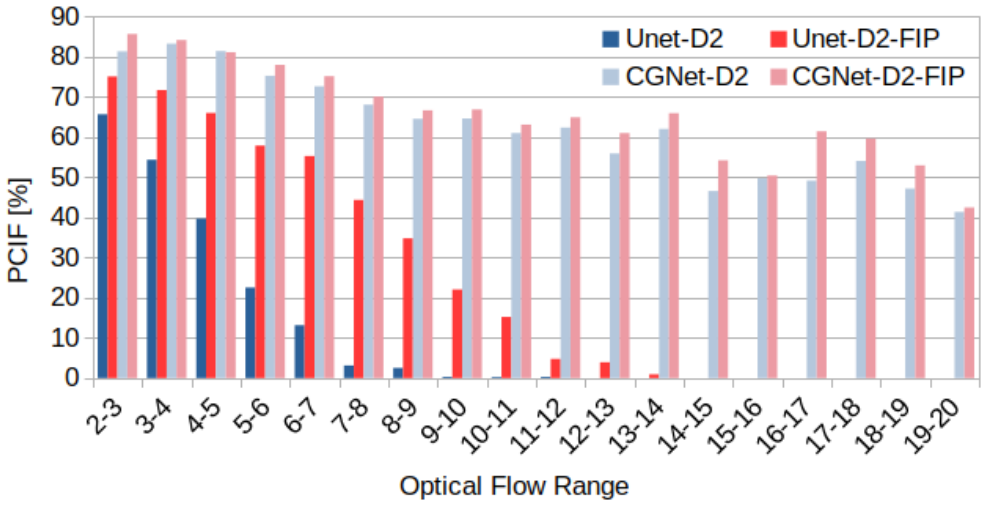


Figure 3.12: PCIF at threshold $r_1 = 2$ for Unet-D2 and CGnet-D2 across different motion ranges.

In the final stage, CGNet-based models are compared with state-of-the-art video denoisers, the results are presented in Tables 3.7 and 3.8. Results for ASWIN [50] are reported using independent models for each noise level following the approach of the authors of this method. The results presented for other methods are obtained based on a single model trained to handle videos with varying noise levels. ShiftNet-S [48] and VRT [34] are configured with 4-frame subsequences to ensure the same frame latency as the CGNet-D3 model. VRT* [34] and RVRT* [35] denote the compressed versions introduced in [223]. For Gaussian video denoising, within the small and medium model categories (grouped by GMACs), the architectures trained with FIP consistently outperform alternative approaches. To compare against the largest models, CGNet is extended to a larger variant (i.e., CGNet-L). While this network is competitive, it does not surpass FloRNN [236], and for such a large backbone, incorporating FIP provides only marginal improvements. On real-noise datasets (CRVD and ReCRVD), however, the proposed method (CGNet-D3-FIP) outperforms all other the approaches while being more efficient in terms of GMACs. The qualitative results on CRVD (Figure 3.14), including both indoor and outdoor

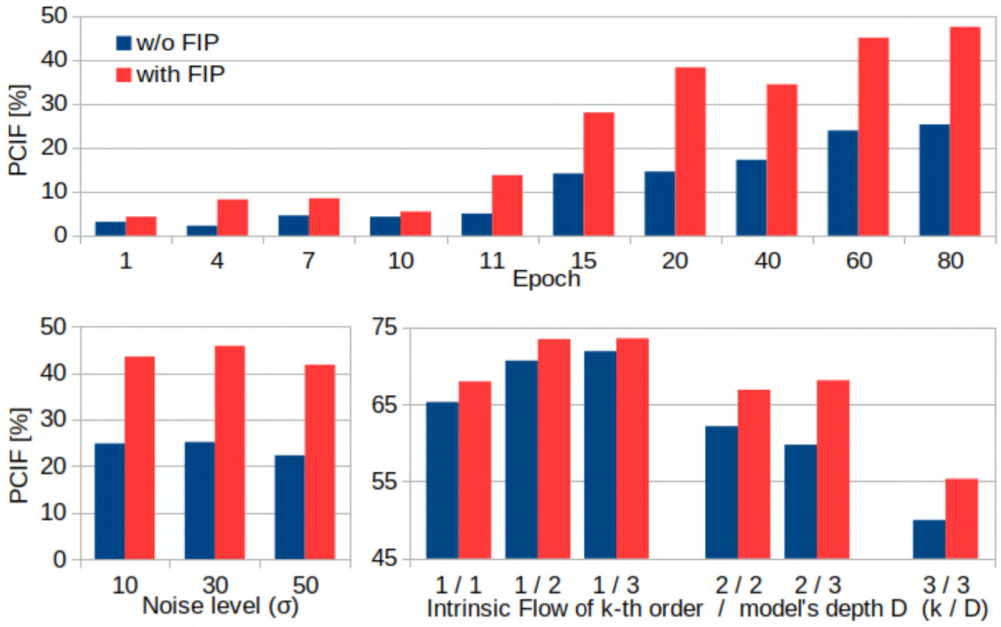


Figure 3.13: PCIF for Unet-D2 across training epochs (top) and noise levels (bottom left), and CGnet-based models across flow orders and depths D (bottom right).

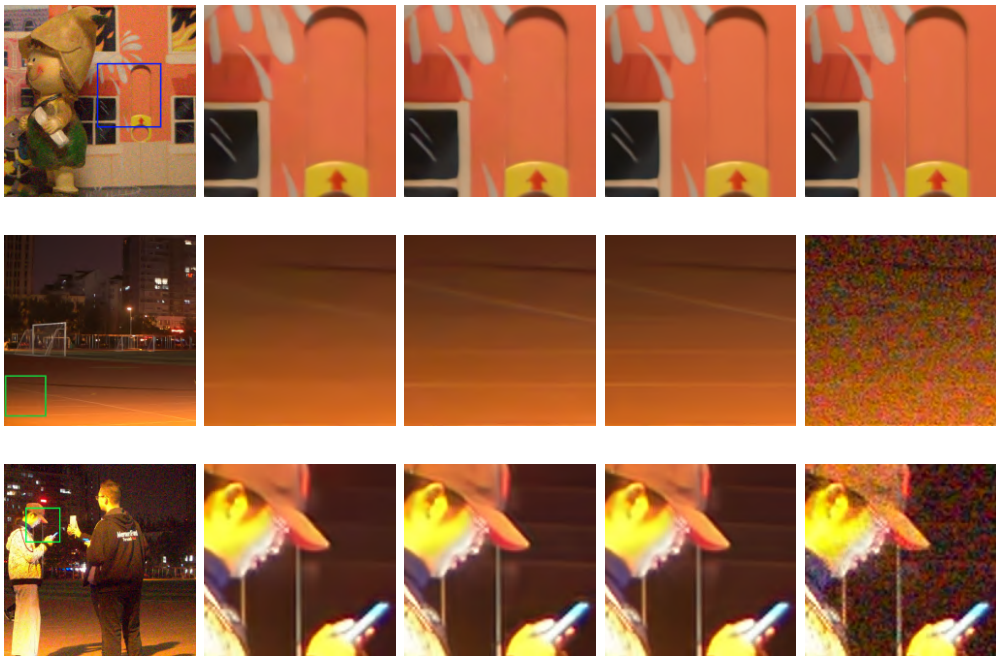
scenes, further support these findings: FloRNN [236] and CGNet-D3-FIP preserve more details after denoising compared to other methods, in line with the quantitative results.

Table 3.7: Comparison of CGNet-based models with state-of-the-art methods for Gaussian video denoising.

Model Size	Video Denoiser	Frame Latency↓	GMACS↓	FPS↑	PSNR↑ DAVIS	PSNR↑ Set8	E(W)↓ DAVIS	E(W)↓ Set8
Small models	BSVD-32 [46]	16	72.5	30.00	35.25	33.09	0.0152	0.0262
	CGNet-D2	2	100.0	6.51	35.63	33.33	0.0139	0.0253
	CGNet-D2-FIP	2	100.0	6.51	**35.79**	**33.39**	**0.0137**	**0.0250**
Medium models	FastDVDnet [45]	2	163.8	22.48	34.87	32.77	0.0162	0.0274
	BSVD-64 [46]	16	291.0	12.10	35.72	33.40	0.0141	0.0253
	ShiftNet-S [48]	3	179.1	3.80	35.66	33.31	0.0138	0.0251
	ASWIN [50]	5	220.1	2.14	35.99	33.53	0.0143	0.0247
	CGNet-D3	3	150.0	4.34	35.80	33.46	0.0136	0.0248
	CGNet-D3-FIP	3	150.0	4.34	**36.00**	**33.55**	**0.0132**	**0.0245**
Large models	EDVR [234]	2	1651.2	1.73	35.82	33.49	0.0139	0.0248
	VRT [34]	3	2209.8	0.38	36.25	33.86	0.1280	0.0241
	FloRNN [236]	3	709.5	4.90	**36.65**	**34.05**	**0.0122**	**0.0234**
	CGNet-L-D3	3	1245.7	0.93	36.41	33.81	0.0126	0.0238
	CGNet-L-D3-FIP	3	1245.7	0.93	36.44	33.81	0.0125	0.0236

Table 3.8: Comparison of CGNet-D3-FIP with state-of-the-art methods on real-world noise raw datasets.

Video Denoiser	Frame Latency↓	GMACS↓	FPS↑	PSNR↑
	CRVD			
RViDeNet [143]	1	2071.6	0.128	44.67
FloRNN [236]	3	2317.0	2.26	45.84
CGNet-D3	3	318.3	2.06	45.78
CGNet-D3-FIP	3	318.3	2.06	**45.88**
	ReCRVD			
RViDeNet [143]	1	2071.6	0.128	43.71
FloRNN [236]	3	2317.0	2.26	44.01
RViDeformer-M [223]	*offline*	143.7	0.32	43.98
RViDeformer-L [223]	*offline*	1790.9	0.23	44.37
RVRT* [35]	1	3152.2	–	43.97
VRT* [34]	*offline*	3056.5	–	44.07
ShiftNet-S[48]	3	378.0	1.59	44.43
CGNet-D3	3	318.3	2.06	44.54
CGNet-D3-FIP	3	318.3	2.06	**44.61**



Noisy RViDeNet [143] FloRNN [236] CGNet-D3-FIP GT-row 1 noisy-row 2,3

Figure 3.14: Qualitative results of real-noise removal performance on video sequences from the CRVD dataset.

3.4 Discussion and Conclusion

In this chapter, the impact of using FIP on implicit motion estimation is studied. It is shown, using novel measures (i.e., PCIF, SFD) and a visualisation technique, that across various architectures FIP improves the utilisation of temporal information. This improvement further translates into better per-frame denoising quality and enhanced temporal consistency. The proposed technique is robust not only across model architectures, but also across noise types, as it provides improvement for both artificial Gaussian and real noise. This chapter proves the first hypothesis, i.e., reducing the information obtained from the central frame in the initial stage of the training, specifically by training the model with the proxy task of frame interpolation, can force the network to put greater emphasis on the neighbouring frames and utilise them to a greater extend. Most of the results presented in this chapter have been published in *IEEE Transactions On Circuits and Systems For Video Technology* [246].

Chapter 4

Low-Resolution Optical Flow and Noise Imbalancing for Efficient Video Denoising

4.1 Introduction

In this chapter, the second component of the first hypothesis and the second hypothesis are both addressed. Optical flow is a widely used tool in video restoration, primarily for aligning frames and enhancing utilisation of the temporal context. However, the optical flow has not been commonly integrated into efficient video denoisers due to high computational costs of both traditional and deep learning–based optical flow estimators, which conflicts with the efficiency requirements of these models. To overcome this limitation, computing optical flow on radically downsized images is proposed, thereby minimising the computational overhead. The hypothesis is that even with such aggressive downsizing, the resulting motion map, hereafter referred to as Low-Resolution Optical Flow (LROF), can still capture sufficient motion information to significantly enhance the video denoising quality. The previously proposed FIP shows a positive impact on video denoising. Nevertheless, it introduces complications such as input frame swapping and Post-FIP intermediate epoch. To overcome these issues, NI is proposed, i.e., an augmentation technique that magnifies noise in the reference frame while reducing it in the remaining ones. Noise levels in all the frames are smoothly balanced throughout the initial training epochs.

4.2 Method

Large motions, which often occur in high-resolution videos, are difficult to handle implicitly for neural networks. To simplify this task, neighbouring frames are often aligned with respect to the reference frame x_t

$$X_t^{\mathcal{W}} = \{\mathcal{W}(x_{t-k}, x_t), \ldots, x_t, \ldots, \mathcal{W}(x_{t+l}, x_t)\} \tag{4.1}$$

where $\mathcal{W}(x_{i1}, x_{i2})$ denotes the alignment of x_{i1} with respect to frame x_{i2}. A set constructed this way can be used instead of X_t for video denoising

$$\tilde{y}_t = x_t - N(X_t^{\mathcal{W}}, m). \tag{4.2}$$

A widely adopted approach for such an alignment relies on warping guided by optical flow

$$\mathcal{W}(x_{i1}, x_{i2}) = \mathcal{W}_{OF}(x_{i1}, OF(x_{i1}, x_{i2})) \tag{4.3}$$

where $OF(x_{i1}, x_{i2})$ denotes the optical flow from frame x_{i2} to frame x_{i1}, and $\mathcal{W}_{\text{OF}}$ warps the frame in accordance with the optical flow.

To align frames more efficiently, it is proposed to compute optical flow on frames downsized by a factor s

$$\mathcal{W}_s(x_{i1}, x_{i2}) = \mathcal{W}_{\mathcal{OF}}(x_{i1}, s * OF(h(x_{i1}) \downarrow_s, h(x_{i2}) \downarrow_s) \uparrow_s) \tag{4.4}$$

where $\downarrow_s$ and $\uparrow_s$ represent, respectively, downsizing and upsampling by the factor s. The computed optical flow is additionally adjusted by scaling its values by the factor of s to account for the change in resolution. The pre-processing function h can be applied to enable or improve the optical flow estimation between the downsized inputs. Its role is to project the input into the format required by the estimator (e.g., most optical flow models expect three-channel inputs) and to reduce the gap between the distribution of the inputs and the data on which the estimators were trained.

For the datasets where the majority of scenes are static, it is particularly important not to introduce motion through warping where none occurs. To achieve it, optical flow vectors with magnitudes smaller than the threshold ρ are filtered out

$$\overline{OF}_{hw} = \begin{cases} 0 & \text{if } |OF_{hw}| < \rho \\ OF_{hw} & \text{if } |OF_{hw}| \geq \rho \end{cases} \tag{4.5}$$

where h and w are the spatial locations (i.e., height and width) of the optical flow vector.

RAW data adaptation RAW images generally follow the Bayer pattern, meaning that each pixel records only a single colour channel. For easier processing, the input sequence X is first reshaped into $X' \in [0, +\infty)^{T \times H/b_f \times W/b_f \times b_f^2}$ using a pixel unshuffle operation, where b_f (which usually equals 2) is the size of the Bayer filter.

To counteract greenish tint and low-brightness characteristics of RAW images, a gamma correction is applied before the optical flow estimation. Additionally, from the total ${b_f}^2$ channels three channels which represent red, green and blue colours

are chosen to match the input expected by the standard optical flow estimators. Specifically, the following h is used in (4.4)

$$h^{RAW}(X') = \begin{bmatrix} {X'_R}^{1/2} & {X'_{G1}}^{1/1.5} & {X'_B}^{1/2} \end{bmatrix} \tag{4.6}$$

where $X'_\kappa \in [0,1)^{T\times H/b\times W/b\times 1}$ represents the clamped time series of channel $\kappa \in \{R, G1, B\}$.

The alignment with LROF is visualised in Figure 4.1. It includes gamma correction applied to RAW data.

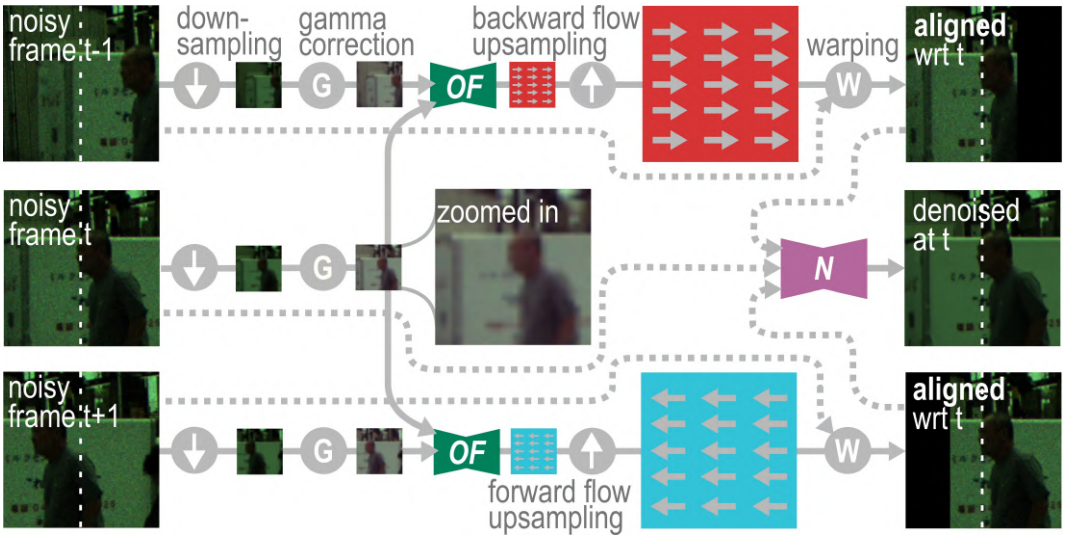


Figure 4.1: Visualisation of alignment with LROF.

Noise Imbalancing To improve the temporal context utilisation, NI is proposed. It relies on the amplification of noise in the central frame and its reduction in adjacent frames during the early stage of the training. Initially, the noise is *heavily imbalanced* between the input frames, with the central frame strongly corrupted and the adjacent frames almost noiseless. Over the course of the pre-training epochs, the noise distribution is gradually rebalanced. At the end of this stage, all the frames receive equal noise levels. This is formalised as

$$\overline{X}_t(\Gamma(f)) = \Pi_{[C_L,C_U]}(Y_t + \Gamma(f) \circ (X_t - Y_t)]) \tag{4.7}$$

where $\circ$ denotes element-wise multiplication, and C_L, C_U denotes the minimal and maximum value of the clipping operation. Afterwards, $Y_t = \{y_{t-k}, ..., y_t, ..., y_{t+j}\}$, $\Gamma(f) = \{\gamma_{-k}, ..., \gamma_0, ..., \gamma_j\}$, $\gamma_0 = f$ and $\gamma_\tau = \frac{1}{f}$ for all $\tau \neq 0$, where f decreases linearly from some $F \in [1, +\infty)$ to 1 throughout the pre-training stage. A visualisation of NI is presented in Figure 4.2.

Combining LROF with NI During the initial epochs, when both alignment and NI are applied, optical flow is computed on the frames without the proposed augmentation and used to warp the frames with imbalanced noise. It is assumed

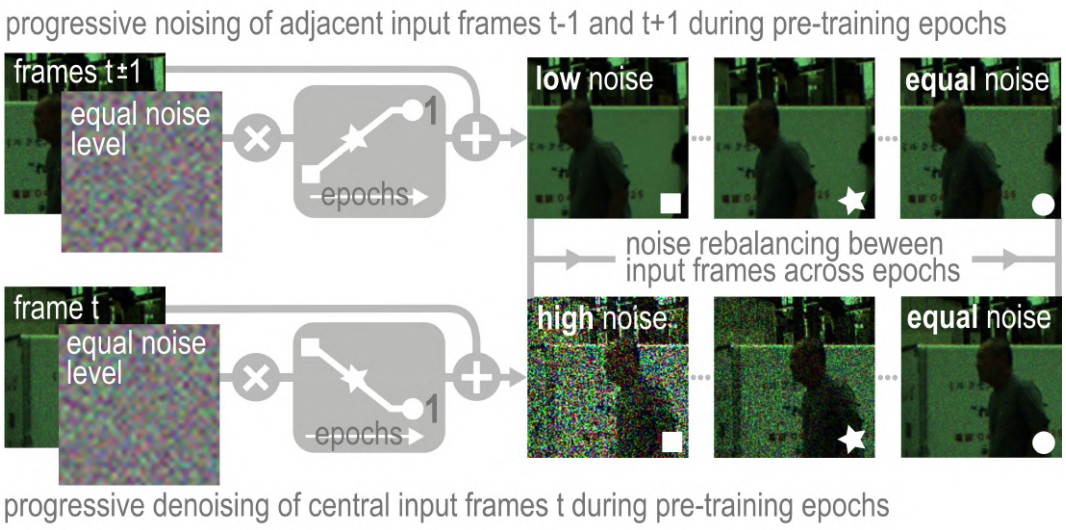


Figure 4.2: Visualisation of NI.

that optical flow computed on augmented frames would be of lower quality than the optical flow obtained from the frames with balanced noise. The objective is to encourage the model to utilise information from neighbouring frames, therefore, the alignment during the pre-training should be as accurate as possible.

4.3 Experiments

Datasets In this chapter, the emphasis is on RAW sensor data corrupted by real-world noise. The primary evaluation is conducted on the recently introduced ReCRVD dataset, which provides relatively long video sequences with realistic motion patterns, making it well-suited to study the proposed techniques. Additionally, experiments on the widely used CRVD dataset are included. However, CRVD is not considered a primary benchmark due to its notable limitations. Specifically, it predominantly features static backgrounds with large foreground object motion and contains only very short sequences, which restricts its applicability to evaluate real-world video denoising performance. Finally, the results on ReCRVD clean frames with artificially added Poisson-Gaussian noise simulating higher ISO levels are also presented.

Evaluation metrics PSNR is employed to assess the denoising quality of the video denoisers. The models are additionally compared in terms of efficiency using FPS computed on an NVIDIA GPU Quadro RTX 5000, GMACs, frame latency, and a number of parameters. Finally, PCIF is used to evaluate the temporal context utilisation. Optical flow computed with the RAFT model [244] on gamma-corrected clean frames serves as the ground-truth optical flow.

Training details The Unet architecture [45] is adopted as the denoising backbone $\mathcal{N}$. The model is lightweight and architecturally simple, making it well suited

Table 4.1: Impact of optical flow scale and gamma correction on denoising performance on the ReCRVD dataset.

s	PSNR w/o gamma	PSNR w. gamma	FPS	GMACs
–	43.91 (0.06)	–	38.93	84.01
12	43.81 (0.34)	44.11 (0.04)	30.72	86.26
6	43.97 (0.35)	44.24 (0.02)	27.79	93.17
4	43.90 (0.35)	44.26 (0.04)	24.26	104.59
3	44.25 (0.04)	43.93 (0.34)	20.99	120.81
2	44.13 (0.28)	44.06 (0.33)	14.68	166.77
1	44.10 (0.26)	44.24 (0.08)	5.687	415.51

for efficiency-constrained scenarios such as deployment on edge devices. It uses only one future frame during inference, thereby introducing minimal temporal latency. To facilitate the processing of Bayer-patterned data, the model is augmented with pixel unshuffling at the network input and pixel shuffling at the output. The training is performed for 80 epochs, each consisting of 3,072,000 patches of the size 192×192, with a batch size of 512. The learning rate is initialised at 0.001 and reduced by factors of 10 and 100 at the 50th and 60th epochs respectively. The proposed NI augmentation technique is applied during the first 10 epochs. For the optical flow computation, the lightweight SpyNet [247] with pre-trained weights is employed. Downsizing the inputs prior to optical flow estimation is performed using average pooling, and bilinear interpolation is used to upsample computed optical flow. Two warping strategies are evaluated, i.e., bilinear interpolation and nearest-neighbour sampling.

4.3.1 Results

Table 4.1 presents the ablation results of the proposed optical-flow-enhanced model, evaluating the impact of the scale factor used during the optical flow estimation and gamma correction. Across most of the scaling factors (except $s = 2$ and $s = 3$), gamma correction improves the average PSNR, demonstrating its effectiveness in enhancing the denoising performance.

Interestingly, the relationship between the scaling factor and denoising quality is not strictly monotonic. Contrary to the expectations, lower scaling factors do not always yield better results. Specifically, the highest performance among the models without gamma correction is achieved at $s = 3$, while the best results for the models with gamma correction are observed at $s = 4$. It is hypothesised that this counterintuitive outcome may stem from the noise suppression introduced by average pooling during downsizing.

To better understand this phenomenon, visualisations of the computed optical flow maps for different s values, with and without gamma correction, are presented in Figure 4.3. In the first example (odd rows), gamma correction substantially improves the optical flow estimation in low-light regions across all the s values, with

the edges in the vicinity of the motion boundaries appearing significantly sharper. In the second example (even rows), which involves stronger noise corruption, it can be observed that the optical flow computed at higher s values appears less noisy, supporting the earlier hypothesis. The reduction of noise in the optical flow is particularly noticeable in the background, for the optical flow computed both with and without gamma correction.

Furthermore, the models trained without gamma correction exhibit high variability in performance, as indicated by large standard deviations across different training runs. In some cases, the models benefit from motion compensation via warping, whereas in others, the optical flow introduces artefacts and degrades the performance. In contrast, the models utilising channel-wise gamma correction produce consistently stable results across most s values.

Considering both the denoising quality and computational efficiency, the model employing gamma correction and $s = 6$ is selected for further diagnostics. This configuration achieves the performance close to that of the top-performing models while maintaining low computational overhead in terms of GFLOPs and inference speed (FPS) when compared with the baseline Unet without optical flow.

For the selected optical flow estimation method, two warping strategies, i.e., **bilinear interpolation** and **nearest-neighbour sampling**, are evaluated. The latter yields a significantly better result, i.e., **44.24 dB**, compared to the former one, i.e., **44.07 dB**. Bilinear interpolation makes the noise smoothed over input frame and spatially correlated, noise of this type is known to be more difficult to remove than spatially uncorrelated noise. This can be a possible reason for the lower result observed for the bilinear interpolation warping strategy as compared with the nearest-neighbour sampling.

Table 4.2 compares two variants of the proposed NI pre-training (both with $N = 16$, without and with clipping of the augmented frames with $C_L = 0$, $C_U = 1$) against FIP and the baseline model. These techniques are applied to three model variants, i.e., without alignment, with alignment using the optical flow computed without gamma correction, and with alignment using the optical flow computed with gamma correction. PSNR values across five noise levels, averaged over five trainings, are presented with a standard deviation in brackets. The impact of the pre-training techniques on the models without alignment is marginal, probably due to the motion magnitude being too large for the proposed network to handle, even when guided towards the temporal context utilisation by the pre-training methods. Interestingly, the models without gamma correction and pre-trained with one of the compared techniques consistently benefit from alignment, in contrast with the model trained without pre-training and relying on the optical flow obtained without gamma correction. Finally, both FIP and NI slightly (by 0.03 dB) improve the results of the model using the frames warped with the optical flow computed after gamma correction, yielding the best-performing solution.

Table 4.3 presents the PCIF results on the ReCRVD dataset. PCIF is reported

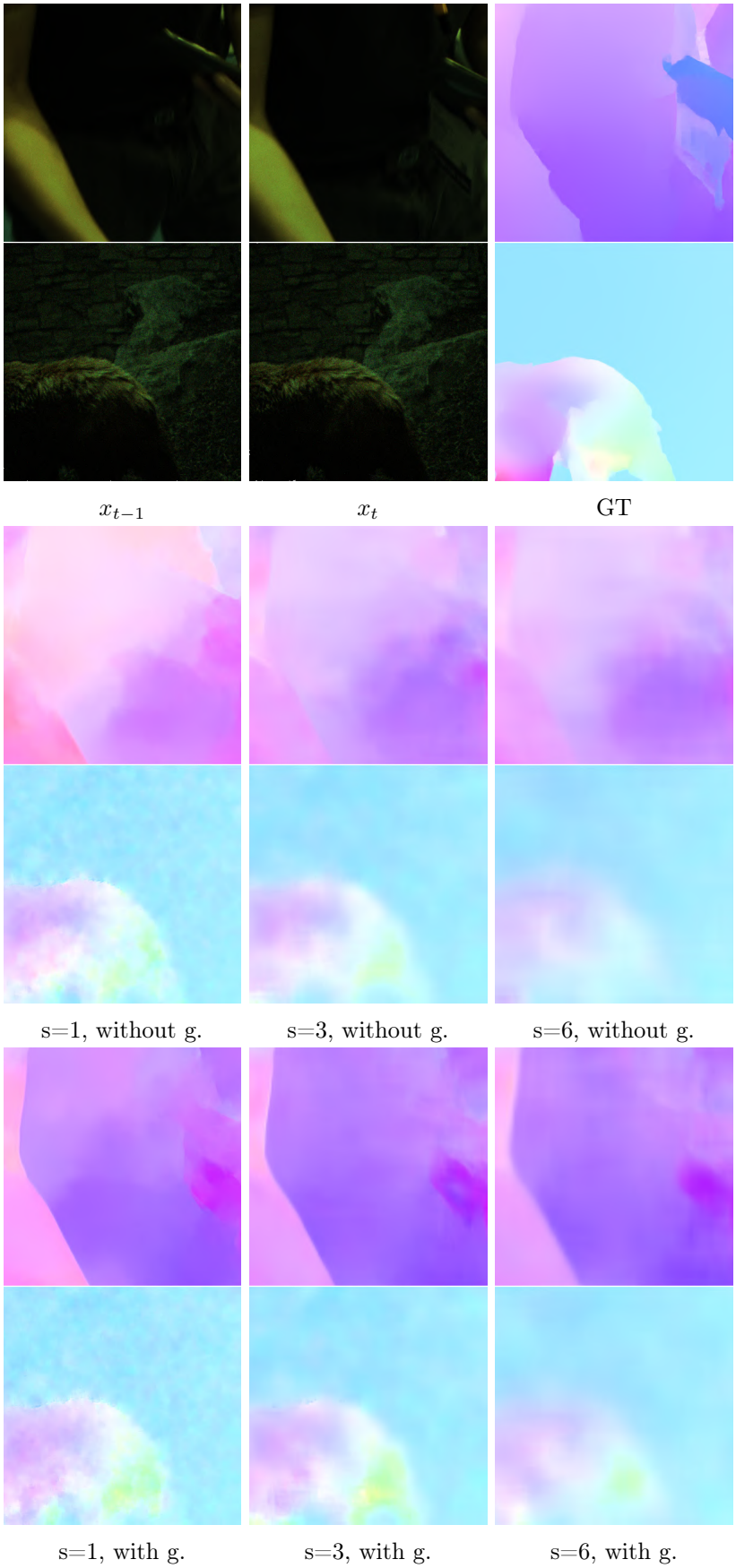


Figure 4.3: Qualitative results of optical flow estimation depending on scaling factor s and employing gamma correction (g.).

Table 4.2: Impact of pre-training variants on denoising performance on the ReCRVD dataset.

FIP	F / clip	w/o OF	with OF w/o gamma	with OF with gamma
✗	✗ / ✗	43.91 (0.06)	43.97 (0.35)	44.24 (0.02)
✓	✗ / ✗	43.91 (0.04)	44.20 (0.04)	44.27 (0.04)
✗	16 / ✗	43.93 (0.05)	44.27 (0.02)	44.02 (0.32)
✗	16 / ✓	43.92 (0.06)	44.22 (0.06)	44.27 (0.05)

across three motion ranges, i.e., small (2-5), medium (5-20), and large (greater than 20). The baseline Unet handles only a portion of the smallest motions. Although augmentation improves performance slightly, it remains ineffective for larger motions. In contrast, the optical flow based model performs significantly better across all the motion levels. Although PCIF decreases with the motion magnitude, it remains robust even for the largest displacements. The difference between the models trained with and without the proposed augmentations is minimal. For reference, PCIF-like values computed using SpyNet optical flows are also reported. These are slightly higher than the corresponding PCIF values. Notably, in some cases, strictly following the ground-truth optical flow may be suboptimal, e.g., in occluded regions or where multiple neighbouring pixels are similarly plausible motion targets.

Table 4.3: Evaluation of PCIF results on ReCRVD with respect to the use of Low-Resolution Optical Flow alignment and NI.

s/ NI	PCIF: r1, r2,r3		
	2,2,5	2,5,20	2,20,∞
✗/ ✗	0.134	0.002	0.000
✗/ ✓	0.183	0.009	0.000
s=6 / ✗	0.675	0.540	0.404
s=6 / ✓	0.673	0.535	0.403
SpyNet, s=6	0.721	0.564	0.430

Tables 4.4 and 4.5 present PSNR values across different noise levels. Table 4.4 presents the results for ISO levels occurring in ReCRVD. For both augmentation and optical flow incorporation, the performance improvements are positively correlated with noise intensity. Specifically, the PSNR gain over the baseline is modest (0.14 dB) at ISO 1600, whereas the improvement for the highest noise level reaches 0.58 dB. This trend aligns with the previous findings indicating that the contribution of temporal context becomes increasingly important as noise severity increases.

Table 4.5 extends this comparison to higher noise levels, simulating ISO up to 409600. To simulate higher ISO, the Poisson-Gaussian model [143] is employed. The scaling parameter of the Poisson noise is calculated using the following formula:

$$\alpha_{(ISO)} = 1.96 * \alpha_{(ISO/2)}. \tag{4.8}$$

Variance of Gaussian noise is analogously computed as

$$\sigma^2{}_{ISO} = 3.76 * \sigma^2{}_{(ISO/2)}. \tag{4.9}$$

In Equations (4.8) and (4.9), the coefficients 1.96 and 3.76 are derived using known Poisson-Gaussian noise parameters corresponding to lower ISO. Across all the simulated noise levels, the combination of the proposed techniques offers significant improvements, ranging from 0.44 db to 0.59 dB. Unlike in the previous experiments, the performance gains observed here do not increase with the noise level. The overall improvements may be partially hindered by the degradation of the optical flow estimation quality under severe noise. Qualitative results (Figure 4.4) on ReCRVD examples degraded by both noise types further confirm the positive impact of the proposed techniques. In the first example, the model enhanced with proposed techniques reconstructs more details of the regular pattern, while in the second it yields sharper results for important elements such as the digits and the face.

Table 4.4: Per ISO evaluation on ReCRVD.

s	**Pre-train**	**1600**	**3200**	**6400**	**12800**	**25600**
✗	✗	46.60	45.71	43.66	41.36	42.21
✗	NI	46.61	45.70	43.70	41.34	42.26
6	✗	46.73	45.84	44.14	41.71	42.75
6	NI	46.74	45.87	44.18	41.76	42.79

Table 4.5: Per ISO evaluation on ReCRVD corrupted with simulated severe noise.

s	**Pre-train**	**25600**	**52000**	**104000**	**208000**	**409600**
✗	✗	42.15	40.81	39.26	37.56	35.62
✗	NI	42.17	40.85	39.33	37.65	35.73
6	✗	42.69	41.38	39.81	38.04	36.00
6	NI	42.71	41.40	39.85	38.08	36.06

Finally, the proposed techniques are evaluated on the CRVD dataset. An unexpected decline in the average PSNR performance is observed when incorporating optical flow. This degradation is attributed to the dataset's predominantly static content, in which the predicted optical flow can be noisy. Furthermore, bilinear upsampling near the motion boundaries may introduce additional artefacts. In this context, warping may cause more harm than benefit. To mitigate these effects, motion filtering that discards flow vectors with magnitudes below a threshold of five pixels is applied. Although this filtering provides an improvement, the most notable gain is achieved by applying the proposed degradation-based augmentation, which increases the PSNR over the baseline by 0.17 dB. Combining both techniques results in the best overall performance.

To further examine the behavior of the models employing the proposed techniques, two additional analyses are performed. First, each frame is partitioned into

Noisy ✗ / ✗ s=6/NI GT

Figure 4.4: Qualitative ablation on ReCRVD dataset; **top:** real noise at ISO25600, **bottom:** synthetic noise at ISO208000.

a background region (areas where optical flow vectors have a magnitude of less than five pixels) and a foreground region (areas where optical flow vectors have a magnitude of five or more pixels), and the PSNR is computed separately for each part. The results indicate that models incorporating optical flow improve their denoising performance in dynamic foreground areas, albeit with a slight performance drop within static regions. This represents a favourable trade-off, as dynamic regions are typically more challenging to denoise, In fact, for the baseline model, PSNR for background pixels is nearly 2 dB higher than for the foreground region.

Second artificial motion is introduced. It is obtained by shifting the frames $t-1$ and $t+1$ diagonally (upwards-left and downwards-right) by M_A pixels to simulate camera motion. To prevent border effects, PSNR is computed on a centre-cropped region (64 pixels removed from each border). The results indicate that the models without optical flow are significantly affected even by slight artificial motion, whereas the models incorporating optical flow demonstrate greater robustness. The results described above are presented in Table 4.6.

Table 4.6: Ablation study results on CRVD.

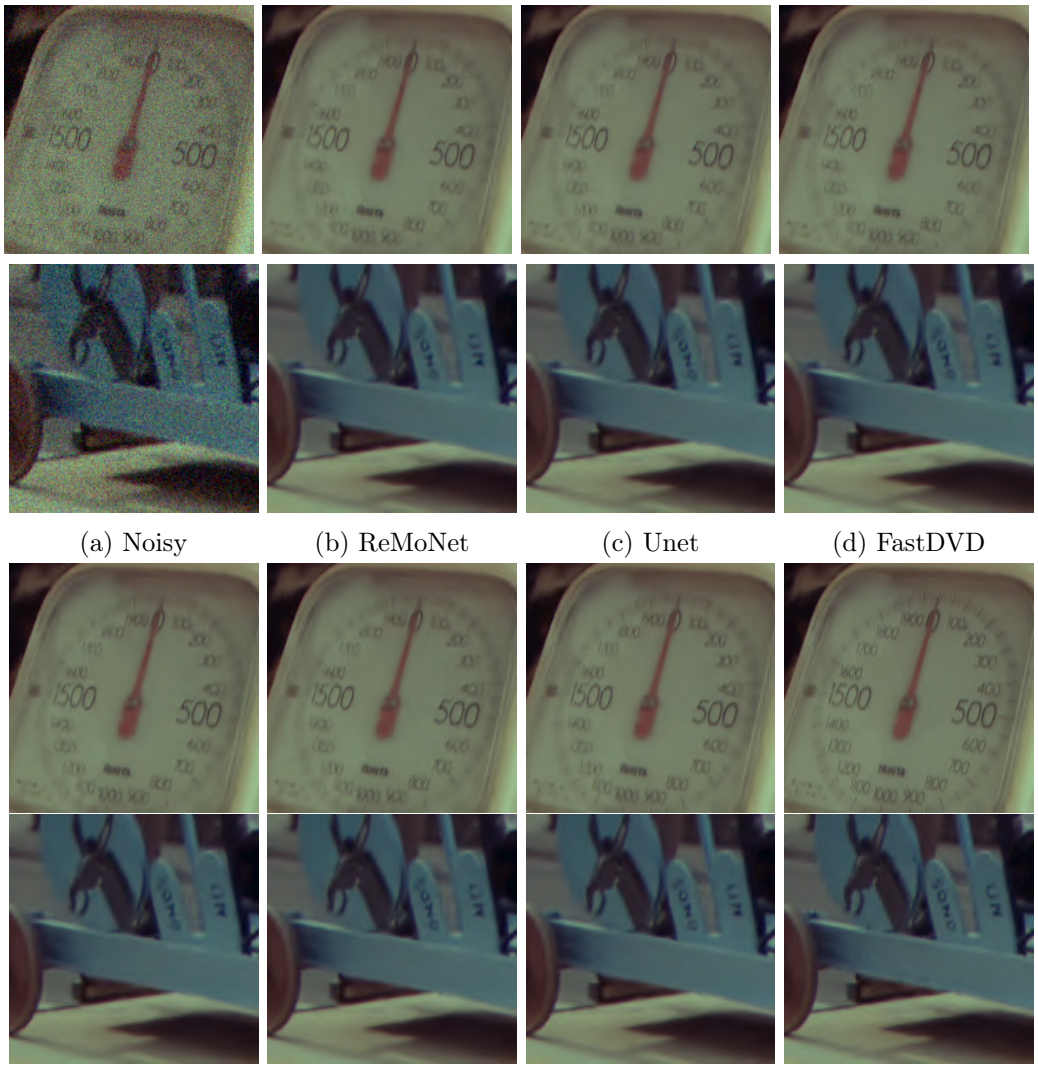
OF	**pre-train**	**PSNR**	**PSNR**		**PSNR for M_A=**			
			Fg	**Bg**	0	2	6	10
x	x	44.04	43.31	45.13	43.75	43.14	43.05	43.05
s=6	x	43.60	43.03	44.48	43.34	43.33	43.29	43.30
s=6, $\rho = 5$	x	43.69	42.97	44.71	43.41	42.91	42.80	43.19
x	FIP	44.18	43.47	45.26	43.90	43.28	43.20	43.20
x	NI	44.21	43.47	45.31	43.92	43.29	43.19	43.18
s=6, $\rho = 5$	FIP	44.18	43.51	45.19	43.90	43.39	43.32	43.67
s=6, $\rho = 5$	NI	44.24	43.55	45.28	43.96	43.42	43.33	43.74

Table 4.7: Comparison of LVD with state-of-the-art methods on ReCRVD.

Method	**PSNR**	**FPS**	**frame latency**	**GMACs**	**params**
Unet	43.91	38.93	1	84.01	1.24
FastDVDnet [45]	43.94	7.34	2	336.03	2.48
BSVD-c32 [46]	44.11	15.20	16	154.36	2.46
BSVD-c64 [46]	44.24	4.43	16	614.51	9.82
ReMoNet [158]	43.70	40.78	4	29.57	0.81
LVD	44.27	27.79	1	93.17	2.20

The method incorporating both proposed techniques, i.e., NI and LROF, is referred as Lightweight Video Denoiser (LVD). By default, it employs $s = 6$ and $N = 16$. The model is compared with state-of-the-art approaches in efficient video denoising on the ReCRVD dataset (Table 4.7). To train the models, the same patch size (192x192) is used, and pixel unshuffling and shuffling are incorporated respectively before and after the network to facilitate processing Bayer-patterned data. Among the evaluated approaches, BSVD-c64 [46] is the only method that reaches a level of denoising performance comparable to LVD. Nevertheless, this comes at a considerable efficiency cost, i.e., it operates at approximately seven times lower FPS, introduces temporal latency of 16 frames, and requires over 4.5 times as many parameters. The lighter variant, i.e., BSVD-c32 [46], represents the second-best option in terms of quality, yet it remains significantly less practical, being roughly twice as slow as LVD while still incurring the same 16-frame latency. In contrast, the methods closer to the proposed model in terms of speed, such as baseline Unet and ReMoNet [158], demonstrate noticeably lower PSNR scores. These findings indicate that LVD provides a more favorable balance between efficiency and denoising quality, making it particularly suitable for the scenarios where both computational cost and restoration accuracy are of primary importance. Additionally, qualitative comparisons of the evaluated methods are presented in Figure 4.5. The visual results demonstrate that the proposed approach preserves fine details more effectively, e.g., the sharpness of letters and numbers, while simultaneously producing smooth outputs in textureless regions.

In Table 4.8, a comparison between LVD and state-of-the-art denoisers on CRVD dataset is presented, all referenced results are taken from [47]. The proposed model achieves results comparable to FastDVDnet [45] while requiring less than one-third of the GMACs. Additionally, a smaller version of LVD is trained with approximately 1.5 times smaller number of channels in convolution layers and $s = 12$. This reduced model achieves the results on par with EMVD [47], a highly efficient model designed to perform well on predominantly static data such as CRVD.



(a) Noisy (b) ReMoNet (c) Unet (d) FastDVD

(e) BSVD32 (f) BSVD64 (g) LVD (h) GT

Figure 4.5: Qualitative results for state-of-the-art methods on the ReCRVD dataset.

Table 4.8: Comparison of LVD with state-of-the-art methods on CRVD.

Method	**PSNR**	**GMACs**	**Frame latency**
EDVR [234]	44.71	1544.49	2
RViDeNet [143]	44.08	982.52	1
FastDVDnet [45]	44.30	332.50	2
EMVD [47]	44.05	39.76	0
LVD	44.24	93.17	1
LVD^{small}	44.02	39.64	1

4.3.2 Deployment

Given that the proposed solution is developed with efficiency constraints in mind, its performance is additionally evaluated on alternative deployment platforms, complementing the GPU-based evaluation presented in the previous section. The obtained results are presented in Tables 4.9, 4.10 and 4.11. They include the outcomes for the model with the proposed frame alignment trained with NI augmentation and basic Unet for comparison. For each configuration, the model exhibiting median performance among the trained realisations is selected.

Firstly, PSNR and FPS achieved on Intel® Core™ i9-10900X central processing unit are presented in Table 4.9. For the deployment, OpenVINO toolkit is used. Two variants are evaluated, i.e., the one maintaining all the layers in full-precision (FP32) and another employing INT8 quantisation for all the layers except the first and the last convolutional layers of Unet. While quantisation leads to a noticeable reduction in PSNR, the resulting acceleration in inference speed provides a favorable trade-off between accuracy and computational efficiency. It is worth noting that the model trained on the CRVD dataset exhibits greater sensitivity to quantisation, resulting in a larger drop in PSNR as compared with the model trained on ReCRVD.

Table 4.9: Deployment results for Intel® Core™ i9-10900X.

	ReCRVD	CRVD	FPS	ReCRVD	CRVD	FPS
	Intel i9, w/o quant			**Intel i9, with quant**		
Unet	43.92	44.06	7.87	43.61	43.32	17.81
LVD	44.27	44.26	6.85	43.86	43.49	14.46

Next, deployment results for the NVIDIA Jetson AGX Orin are presented in Table 4.10. Models are compiled into a TensorRT representation using Torch JIT to improve efficiency. In all the tests, the maximum performance mode (MAXN) is employed. All the model operations are executed on the GPU to avoid data transfers and execution of operations on CPU, which would otherwise slow down the inference. The models are evaluated using FP16 and FP32 precisions. The version utilising lower precision provides a substantial gain in FPS with a minimal impact on the denoising quality.

Table 4.10: Deployment results for Jetson AGX Orin.

	ReCRVD	CRVD	FPS	ReCRVD	CRVD	FPS
	Jetson FP16			**Jetson FP32**		
Unet	43.92	44.06	31.38	43.91	44.02	47.45
LVD	44.28	44.25	27.19	44.26	44.24	41.03

Finally, the results for the models deployed on two devices with Qualcomm accelerators are presented in Table 4.11. The first platform is the Samsung Galaxy S24 with a Qualcomm Snapdragon microprocessor, and the second is the Qualcomm Dragonwing QCS6490. For the deployment, QNN context binary runtime is em-

ployed, and the Qualcomm Hexagon Tensor Processor, a Neural Processing Unit, is leveraged. Similar to the deployment on the Jetson platform, it is confirmed that all the operations in the model graph of the tested models are fully supported by the NPU. Because NPU does not natively support FP32 precision, only FP16 precision is used. The tests are performed using the maximum performance mode (burst).

It can be observed that incorporating optical flow estimation significantly impacts the efficiency on the Samsung Galaxy S24. For the naively deployed model, the primary bottleneck is the downsizing frames prior to optical flow estimation. The initial implementation without any optimisation reaches only 4.48 FPS. Dividing frames into 6×6 patches and stacking them along the batch dimension before downsizing is essential for performance and does not affect the optical flow quality. The model with this modification is nearly 3.5 times faster, increasing the performance to 15.50 FPS. Similarly, the upsampling of obtained optical flow can also be carried out in a patch-wise manner. Unlike patch-wise downsizing, this approach slightly affects the quality of the estimated optical flow. However, the experiments show that for 5×5 patches the impact on the final PSNR is marginal, an approximately 0.01 dB drop in PSNR is observed. Patch-wise upsampling, combined with performing some consecutive operations in a patch-wise manner, significantly boosts the inference speed to 18.60 FPS. It is also observed that processing images in large patches can substantially improve the performance. For instance, splitting the input along the width dimension into two halves, processing them separately, and then concatenating the outputs increases the throughput to 28.65 FPS during inference. Again, only a minor performance drop is observed, less than 0.01 dB in PSNR. The reported FPS for the baseline Unet includes the last proposed optimisation, without which the model reaches 39.2 FPS. The same modifications are also applied on the Qualcomm Dragonwing QCS6490, improving its efficiency in a similar manner.

Table 4.11: Deployment results for Samsung Galaxy S24 and Qualcomm Dragonwing QCS6490.

	ReCRVD	CRVD	FPS	ReCRVD	CRVD	FPS
	Samsung S24			**Dragonwing QCS6490**		
Unet	43.91	44.03	40.32	43.91	44.03	30.48
LVD	44.25	44.25	28.65	44.25	44.25	20.75

It can be observed that on platforms based on the Qualcomm processor as well as on the initially used GPU, the impact of incorporating optical flow warping on FPS is not directly proportional to the corresponding increase in GMACs. One contributing factor is a relatively high computational cost of the warping operation itself which requires accessing data from memory in a non-sequential manner, thereby making the process time-consuming. Furthermore, unlike convolutions or common activation functions, warping is a less frequently used operation and, consequently, not as extensively optimised as elementary deep neural network components.

4.4 Discussion and Conclusion

In this chapter, the impact of using LROF and NI on video denoising quality is studied. It is demonstrated that optical flow computed on radically downsized frames (up to a factor of twelve) can improve the denoising quality. This effect is particularly pronounced for data with substantial motion, but it also improves robustness for more static content. Contrary, NI is especially important for predominantly static data. Nevertheless, it is showed that both methods are compatible and their combination brings the best results across datasets. The model incorporating both proposed techniques, referred to as LVD, achieves state-of-the-art results on ReCRVD and demonstrates robust performance under varying motion, which is comparable with state-of-the-art methods on CRVD.

This chapter complements the answer to the first hypothesis: differentiation of the noise amount during the initial training stage, such that the central frame is more heavily corrupted while the neighbouring frames are clearer, also yields better temporal context utilization and higher denoising quality. Furthermore, the chapter addresses the second hypothesis as well. That is, the optical flow computed on radically downsized frames substantially compensates for inter-frame motion and, consequently, significantly improves the video denoising quality. Most of the work presented in this chapter is a part of the manuscript that is after the first submission to *IEEE Signal Processing Letters*.

Chapter 5

Conclusions and Outlook

5.1 Pre-Training

In this dissertation, it is experimentally demonstrated that limiting the information obtained from the denoised frame in the initial stage of the training can lead to improvement of temporal context utilisation. This is especially important for smaller networks which are more prone to relying extravagantly on the central frame. The proposed training techniques improve video denoising quality without compromising the model efficiency.

In Chapter 3, it is shown that FIP leads to better implicit motion estimation resulting in higher per-frame quality and better temporal consistency. To achieve the best results, the proposed FIP is combined with an additional stage called Post-FIP, which steadily transits from the proxy task to the target one.

In Chapter 4, an alternative method for enhancing video denoising through a carefully designed pre-training is introduced. At the beginning of the training, access to the signal from the central frame is also limited. Unlike FIP, the central frame is not masked. Instead, its noise is amplified. Simultaneously, the noise in other frames is diminished to facilitate their utilisation. NI achieves comparable or superior results to FIP, depending on the dataset. At the same time, NI is easier to implement as it does not require an intermediate training stage, in contrast to FIP this information-limiting technique is easily gradable.

5.2 Low-Resolution Optical Flow

In the dissertation it is also demonstrated that LROF, i.e., the optical flow computed on downsized degraded frames, can lead to significant improvements in denoising performance. In particular, employing the optical flow estimated from the frames downsized by a factor of six brings about results on par with a solution using the optical flow computed in the original resolution on a dataset with realistic motion. Further downsizing reduces the improvement, but even heavier, twelvefold,

downsizing still enables significant denoising correction while increasing the number of the required GMACc operations by less than 3%.

5.3 Contributions

To conclude, this dissertation presents the following contributions. First, a novel methodology for training video restoration models is introduced. It is designed to facilitate more effective utilisation of neighbouring frames. The approach encourages the model to depend more on auxiliary frames rather than predominantly on the denoised reference frame by limiting the information available from the latter and facilitating information extraction from the other frames. Two realisations of this methodology are proposed and their effectiveness is validated on one of the most widely studied video restoration tasks, i.e., video denoising. In practical terms, both the proposed methods, i.e., FIP and NI, can be incorporated into the training of efficient video denoising models to enhance their restoration performance without increasing computational costs during inference.

Secondly, the influence of downsizing input frames on the effectiveness of optical flow in supporting video denoising is examined. The presented experiments demonstrate that the optical flow estimated from heavily downsized frames can also yield significant improvements in denoising quality. At the same time, this approach substantially reduces the computational cost of optical flow estimation, providing a favourable balance between efficiency and performance. In efficiency-constrained methods, the proposed technique enables the use of optical flow, which has previously been largely avoided due to its high computational cost.

Finally, a model integrating the two proposed techniques, i.e., LROF and NI, is presented. Despite being based on a compact and relatively simple Unet architecture, the proposed method achieves state-of-the-art performance among the efficient models on the ReCRVD dataset which features real noise and realistic motion. Moreover, on the more commonly used but less realistic CRVD dataset, characterised by real noise but predominantly static scenes, the model attains results on par with other efficient methods that rely on a larger number of input frames. Deployment of the proposed model on edge devices, including Jetson AGX Orin, Samsung Galaxy S24 and Qualcomm Dragonwing QCS649, demonstrates that the proposed model is capable of real-time processing in practical applications while maintaining low frame latency.

5.4 Future Work

The dissertation primarily focuses on efficiency-oriented techniques that improve motion estimation and temporal information utilisation. The presented results validate both hypotheses formulated in the dissertation. A natural extension of this

work involves applying the proposed techniques to other video modalities beyond RGB and RAW data, e.g., hyperspectral, luma, ultrasound, depth or event-based videos, and subsequently to other video restoration tasks, such as video deblurring, super resolution or deraining. The application of LROF appears to be straightforward for most of these tasks. Designing analogous techniques for NI and the progressive noising aspect of FIP for other restoration tasks is more challenging. Both techniques rely on reducing the degradation present in input data (specifically past and future frames) whereas degradations in tasks other than denoising are often not as esaily gradable. This process is more straightforward in training setups where degradation is introduced artificially, e.g., by applying blur with a known kernel, as its parameters are usually adjustable. Because such degradations are not realistic and do not generalise well to the real-world scenarios, they are on the way out. In particular, for super-resolution, it is difficult to smoothly change the degradation level since the scaling ratio is strongly connected with the model architecture.

In more complex architectures, features extracted from the denoised frames serve a dual purpose: they are used both to generate representations important for the denoising process itself and to estimate motion (e.g., in offset computation for deformable convolutions). To perform the second task effectively, which would facilitate the utilisation of information from neighbouring frames, these features should not be strongly degraded. This gives the rise to the need of limiting the information obtained from the central frame in a manner that could be applied more flexibly inside the model. Such an approach could also be easier to implement in the methods that denoise multiple frames concurrently.

Both the proposed pre-training methods rely on heuristically designed schedules for the parameters that regulate access to information from the denoised and auxiliary frames. While these approaches are sufficient to validate the first proposed hypothesis, further improvements could be achieved with a more analytical strategy. This would require the development of additional efficient and comprehensive tools for assessing a model's capability in implicit motion estimation.

The domain gap between the RGB and RAW data is effectively narrowed by applying gamma correction. However, potential benefits of using LROF could be further increased by fine-tuning the optical flow estimation network on the target dataset. To mitigate the negative effect of interpolation near the motion boundaries caused by bilinear interpolation, an additional efficient module could be trained to upsample optical flow more precisely. Finally, the optical flow estimator could be fine-tuned jointly with the denoising model to provide task-oriented optical flow, and LROF could be incorporated in more sophisticated modules such as Flow-Guided Deformable Alignment.

Publications

The author has contributed to the following peer-reviewed journal and conference publications during the Ph.D. studies:

- **Kopa Ostrowski, P.**, Jezierska, A., Grzymkowski, Ł., Stefański, T., & Węsierski, D. A Lightweight RAW Video Denoiser with Coarse Alignment and Imbalanced Noise Pre-training. After the first submission to IEEE Signal Processing Letters. [MNiSW **100 pts**]

- **Kopa Ostrowski, P.**, Węsierski, D., Jezierska, A., & Stefański, T. (2025). Lifting Deep Image Denoisers to Video with Frame Interpolation Pre-training. IEEE Transactions on Circuits and Systems for Video Technology. [MNiSW **140 pts**]

- Katsaros, E.,**Kopa Ostrowski, P.**, Włódarczak, K., Lewandowska, E., Ruminski, J., Siupka-Mróz, D., Lassmann, Ł., Jezierska, A., & Wesierski, D. (2022). Multi-task Video Enhancement for Dental Interventions. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 177-187). Springer, Cham. [MNiSW **140 pts**]

- Katsaros, E., **Kopa Ostrowski, P.**, Wesierski, D., & Jezierska, A. (2021). Concurrent video denoising and deblurring for dynamic scenes. IEEE Access, 9, 157437-157446. [MNiSW **100 pts**]

- **Kopa Ostrowski, P.**, Katsaros, E., Wesierski, D., & Jezierska, A. (2022). BP-EVD: Forward Block-Output Propagation for Efficient Video Denoising. IEEE Transactions on Image Processing, 31, 3809-3824. [MNiSW **200 pts**]

Datasets

The author has contributed to the following publicly available datasets during the Ph.D. studies:

- Katsaros, E., **Kopa Ostrowski, P.**, Jezierska, A., Lewandowska, E., Ruminski, J., & Wesierski, D. (2022). Vident-lab: a dataset for multi-task video processing of phantom dental scenes (1–) . Gdańsk University of Technology. `https://doi.org/10.34808/1jby-ay90`

- Węsierski, D., Jezierska, A., **Kopa Ostrowski, P.**, Lewandowska, E., Katsaros, E., & Żółtowska, A. (2024). Vident-real: an intra-oral video dataset for multi-task learning (1–) [Dataset]. Gdańsk University of Technology. `https://doi.org/10.34808/vjnh-9c35`

Projects

The author has participated in the following research projects during the Ph.D. studies:

- Development of an innovative device, cooperating with standard dental instruments, for non-invasive, optical, real-time monitoring of mouth interior during dental procedures, November 2020 - September 2022. Funding: National Centre for Research and Development POIR.01.01.01- 00-0076/19
- Wide field high resolution optical inspection of wind turbines with multi-task video processing at the edge, February - August 2023. Funding: IDUB Ventus Hydrogenii Redivivus, Gdańsk University of Technology
- Semantic segmentation of dental structures in video signal, May - August 2023. Funding: IDUB Aurum, Gdańsk University of Technology

Other research activities

Additional research activities conducted during the Ph.D. studies include:

- Poster presentation at the 25th International Conference on Medical Image Computing and Computer-Assisted Intervention, September 2022,
- Research internship at the Centre for Visual Computing, Paris-Saclay, CentraleSupélec, September - December 2023.

List of Figures

List of Tables

Bibliography

[1] P. Chatterjee and P. Milanfar, "Is denoising dead?," *IEEE transactions on image processing*, vol. 19, no. 4, pp. 895–911, 2009.

[2] A. Levin and B. Nadler, "Natural image denoising: Optimality and inherent bounds," in *Computer vision and pattern recognition conference*, pp. 2833–2840, IEEE, 2011.

[3] A. Levin, B. Nadler, F. Durand, and W. T. Freeman, "Patch complexity, finite pixel correlations and optimal denoising," in *European conference on computer vision*, pp. 73–86, Springer, 2012.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[5] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[6] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[7] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[8] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[9] A. de Brebisson and G. Montana, "Deep neural networks for anatomical brain segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 20–28, 2015.

[10] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2147–2154, 2014.

[11] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," *Advances in neural information processing systems*, vol. 26, 2013.

[12] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[13] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[14] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[15] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[16] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[17] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, *et al.*, "Grandmaster level in starcraft ii using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[18] R. McCarthy, F. R. Sanchez, Q. Wang, D. C. Bulens, K. McGuinness, N. O'Connor, and S. J. Redmond, "Solving the real robot challenge using deep reinforcement learning," *arXiv preprint arXiv:2109.15233*, 2021.

[19] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. Scaramuzza, "Champion-level drone racing using deep reinforcement learning," *Nature*, vol. 620, no. 7976, pp. 982–987, 2023.

[20] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 4960–4964, IEEE, 2016.

[21] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE international*

conference on acoustics, speech and signal processing (ICASSP), pp. 4774–4778, IEEE, 2018.

[22] S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, *et al.*, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, vol. 12, p. 1, 2016.

[23] S. Ö. Arık, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman, *et al.*, "Deep voice: Real-time neural text-to-speech," in *International conference on machine learning*, pp. 195–204, PMLR, 2017.

[24] Y. A. Li, C. Han, V. Raghavan, G. Mischler, and N. Mesgarani, "Styletts 2: Towards human-level text-to-speech through style diffusion and adversarial training with large speech language models," *Advances in neural information processing systems*, vol. 36, pp. 19594–19621, 2023.

[25] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *Proceedings of the 10th ACM conference on recommender systems*, pp. 191–198, 2016.

[26] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*, pp. 173–182, 2017.

[27] J. Saxe and K. Berlin, "Deep neural network based malware detection using two dimensional binary program features," in *2015 10th international conference on malicious and unwanted software (MALWARE)*, pp. 11–20, IEEE, 2015.

[28] N. Lu, D. Li, W. Shi, P. Vijayakumar, F. Piccialli, and V. Chang, "An efficient combined deep neural network based malware detection framework in 5g environment," *Computer Networks*, vol. 189, p. 107932, 2021.

[29] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," *Machine learning*, vol. 5, p. 2, 2018.

[30] G. E. Dahl, N. Jaitly, and R. Salakhutdinov, "Multi-task neural networks for qsar predictions," *arXiv preprint arXiv:1406.1231*, 2014.

[31] J. M. Stokes, K. Yang, K. Swanson, W. Jin, A. Cubillos-Ruiz, N. M. Donghia, C. R. MacNair, S. French, L. A. Carfrae, Z. Bloom-Ackermann, *et al.*, "A deep learning approach to antibiotic discovery," *Cell*, vol. 180, no. 4, pp. 688–702, 2020.

[32] 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.

[33] 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," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

[34] J. Liang, J. Cao, Y. Fan, K. Zhang, R. Ranjan, Y. Li, R. Timofte, and L. Van Gool, "Vrt: A video restoration transformer," *IEEE transactions on image processing*, vol. 33, pp. 2171–2182, 2024.

[35] J. Liang, Y. Fan, X. Xiang, R. Ranjan, E. Ilg, S. Green, J. Cao, K. Zhang, R. Timofte, and L. V. Gool, "Recurrent video restoration transformer with guided deformable attention," *Advances in neural information processing systems*, vol. 35, pp. 378–393, 2022.

[36] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, "Noise2noise: Learning image restoration without clean data," in *International conference on machine learning*, pp. 2965–2974, PMLR, 2018.

[37] J. Batson and L. Royer, "Noise2self: Blind denoising by self-supervision," in *International conference on machine learning*, pp. 524–533, PMLR, 2019.

[38] A. Limsuebchuea, R. Duangsoithong, and P. Phukpattaranont, "Self-augmented noisy image for noise2noise image denoising," *IEEE Access*, vol. 12, pp. 71076–71087, 2024.

[39] A. Krull, T.-O. Buchholz, and F. Jug, "Noise2void-learning denoising from single noisy images," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2129–2137, 2019.

[40] H. Chen, J. Gu, Y. Liu, S. A. Magid, C. Dong, Q. Wang, H. Pfister, and L. Zhu, "Masked image training for generalizable deep image denoising," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1692–1703, 2023.

[41] D. Ryou, I. Ha, H. Yoo, D. Kim, and B. Han, "Robust image denoising through adversarial frequency mixup," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2723–2732, 2024.

[42] A. Karim, E. Ahmed, S. Azam, B. Shanmugam, and P. Ghosh, "Mitigating the latency induced delay in ip telephony through an enhanced de-jitter buffer," in *Mobile Computing and Sustainable Informatics: Proceedings of ICMCSI 2021*, pp. 1–16, Springer, 2021.

[43] E. Chang, H. T. Kim, and B. Yoo, "Virtual reality sickness: a review of causes and measurements," *International journal of human–computer interaction*, vol. 36, no. 17, pp. 1658–1682, 2020.

[44] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," in *International conference on learning representations*, 2019.

[45] M. Tassano, J. Delon, and T. Veit, "Fastdvdnet: Towards real-time deep video denoising without flow estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1354–1363, 2020.

[46] C. Qi, J. Chen, X. Yang, and Q. Chen, "Real-time streaming video denoising with bidirectional buffers," in *Proceedings of the 30th ACM international conference on multimedia*, pp. 2758–2766, 2022.

[47] M. Maggioni, Y. Huang, C. Li, S. Xiao, Z. Fu, and F. Song, "Efficient multi-stage video denoising with recurrent spatio-temporal fusion," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3466–3475, 2021.

[48] D. Li, X. Shi, Y. Zhang, K. C. Cheung, S. See, X. Wang, H. Qin, and H. Li, "A simple baseline for video restoration with grouped spatial-temporal shift," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9822–9832, 2023.

[49] H. Chen, Y. Jin, K. Xu, Y. Chen, and C. Zhu, "Multiframe-to-multiframe network for video denoising," *IEEE transactions on multimedia*, vol. 24, pp. 2164–2178, 2021.

[50] L. Lindner, A. Effland, F. Ilic, T. Pock, and E. Kobler, "Lightweight video denoising using aggregated shifted window attention," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 351–360, 2023.

[51] K. C. Chan, S. Zhou, X. Xu, and C. C. Loy, "On the generalization of basicvsr++ to video deblurring and denoising," *arXiv preprint arXiv:2204.05308*, 2022.

[52] M. Song, Y. Zhang, and T. O. Aydın, "Tempformer: Temporally consistent transformer for video denoising," in *European conference on computer vision*, pp. 481–496, Springer, 2022.

[53] Y. Jin, X. Ma, R. Zhang, H. Chen, Y. Gu, P. Ling, and E. Chen, "Masked video pretraining advances real-world video denoising," *IEEE transactions on multimedia*, 2025.

[54] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[55] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

[56] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1314–1324, 2019.

[57] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[58] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[59] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE transactions on intelligent transportation systems*, vol. 19, no. 1, pp. 263–272, 2017.

[60] I. Freeman, L. Roese-Koerner, and A. Kummert, "Effnet: An efficient structure for convolutional neural networks," in *IEEE international conference on image processing*, pp. 6–10, IEEE, 2018.

[61] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6848–6856, 2018.

[62] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

[63] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[64] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "Ghostnet: More features from cheap operations," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1580–1589, 2020.

[65] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

[66] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "Eca-net: Efficient channel attention for deep convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11534–11542, 2020.

[67] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *European conference on computer vision*, pp. 3–19, 2018.

[68] A. Vaswani, P. Ramachandran, A. Srinivas, N. Parmar, B. Hechtman, and J. Shlens, "Scaling local self-attention for parameter efficient visual backbones," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12894–12904, 2021.

[69] J. Ho, N. Kalchbrenner, D. Weissenborn, and T. Salimans, "Axial attention in multidimensional transformers," *arXiv preprint arXiv:1912.12180*, 2019.

[70] Y. Xiong, Z. Zeng, R. Chakraborty, M. Tan, G. Fung, Y. Li, and V. Singh, "Nyströmformer: A nyström-based algorithm for approximating self-attention," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, pp. 14138–14148, 2021.

[71] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[72] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," in *International conference on learning representations*, 2017.

[73] E. Zamfir, Z. Wu, N. Mehta, Y. Zhang, and R. Timofte, "See more details: Efficient image super-resolution by experts mining," in *International conference on machine learning*, 2024.

[74] Y. Han, G. Huang, S. Song, L. Yang, H. Wang, and Y. Wang, "Dynamic neural networks: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 11, pp. 7436–7456, 2021.

[75] X. Ding, Y. Guo, G. Ding, and J. Han, "Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1911–1920, 2019.

[76] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "Repvgg: Making vgg-style convnets great again," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13733–13742, 2021.

[77] H. Chen, Y. Wang, J. Guo, and D. Tao, "Vanillanet: the power of minimalism in deep learning," *Advances in neural information processing systems*, vol. 36, pp. 7050–7064, 2023.

[78] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "Icnet for real-time semantic segmentation on high-resolution images," in *European conference on computer vision*, pp. 405–420, 2018.

[79] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *European conference on computer vision*, pp. 325–341, 2018.

[80] R. P. Poudel, S. Liwicki, and R. Cipolla, "Fast-scnn: Fast semantic segmentation network," *arXiv preprint arXiv:1902.04502*, 2019.

[81] Y. Hong, H. Pan, W. Sun, and Y. Jia, "Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes," *arXiv preprint arXiv:2101.06085*, 2021.

[82] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[83] 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*, pp. 234–241, Springer, 2015.

[84] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *arXiv preprint arXiv:1606.02147*, 2016.

[85] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi, "Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation," in *European conference on computer vision*, pp. 552–568, 2018.

[86] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, pp. 21–37, Springer, 2016.

[87] H. Zeng, J. Cai, L. Li, Z. Cao, and L. Zhang, "Learning image-adaptive 3d lookup tables for high performance photo enhancement in real-time," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 4, pp. 2058–2073, 2020.

[88] C. Liu, H. Yang, J. Fu, and X. Qian, "4d lut: learnable context-aware 4d lookup table for image enhancement," *IEEE transactions on image processing*, vol. 32, pp. 4742–4756, 2023.

[89] B. Zoph and Q. Le, "Neural architecture search with reinforcement learning," in *International conference on learning representations*, 2017.

[90] C. Liu, L.-C. Chen, F. Schroff, H. Adam, W. Hua, A. L. Yuille, and L. Fei-Fei, "Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 82–92, 2019.

[91] 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*, pp. 59–64, IEEE, 2021.

[92] H. Zhang, Y. Li, H. Chen, and C. Shen, "Memory-efficient hierarchical neural architecture search for image denoising," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3657–3666, 2020.

[93] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8697–8710, 2018.

[94] H. Cai, L. Zhu, and S. Han, "Proxylessnas: Direct neural architecture search on target task and hardware," in *International conference on learning representations*, 2019.

[95] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," in *International conference on learning representations*, 2018.

[96] H. Guan, A. Malevich, J. Yang, J. Park, and H. Yuen, "Post-training 4-bit quantization on embedding tables," *CoRR*, 2019.

[97] R. Banner, Y. Nahshan, and D. Soudry, "Post training 4-bit quantization of convolutional networks for rapid-deployment," *Advances in neural information processing systems*, vol. 32, 2019.

[98] F. Li, B. Liu, X. Wang, B. Zhang, and J. Yan, "Ternary weight networks," *arXiv preprint arXiv:1605.04711*, 2016.

[99] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *European conference on computer vision*, pp. 525–542, Springer, 2016.

[100] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," in *International conference on machine learning*, pp. 1737–1746, PMLR, 2015.

[101] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 1389–1397, 2017.

[102] Z. Zhuang, M. Tan, B. Zhuang, J. Liu, Y. Guo, Q. Wu, J. Huang, and J. Zhu, "Discrimination-aware channel pruning for deep neural networks," *Advances in neural information processing systems*, vol. 31, 2018.

[103] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," in *International conference on learning representations*, 2018.

[104] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding," in *ICLR*, 2016.

[105] J. Wu, Y. Wang, Z. Wu, Z. Wang, A. Veeraraghavan, and Y. Lin, "Deep k-means: Re-training and parameter sharing with harder cluster assignments for compressing deep convolutions," in *International conference on machine learning*, pp. 5363–5372, PMLR, 2018.

[106] Y.-D. Kim, E. Park, S. Yoo, T. Choi, L. Yang, and D. Shin, "Compression of deep convolutional neural networks for fast and low power mobile applications," *arXiv preprint arXiv:1511.06530*, 2015.

[107] V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, and V. Lempitsky, "Speeding-up convolutional neural networks using fine-tuned cp-decomposition," *arXiv preprint arXiv:1412.6553*, 2014.

[108] P. Yin, J. Lyu, S. Zhang, S. Osher, Y. Qi, and J. Xin, "Understanding straight-through estimator in training activation quantized neural nets," in *International conference on learning representations*, 2019.

[109] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," *Advances in neural information processing systems*, vol. 28, 2015.

[110] G. Li, C. Qian, C. Jiang, X. Lu, and K. Tang, "Optimization based layer-wise magnitude-based pruning for dnn compression.," in *International joint conference on artificial intelligence*, vol. 330, pp. 2383–2389, 2018.

[111] Y. LeCun, J. Denker, and S. Solla, "Optimal brain damage," *Advances in neural information processing systems*, vol. 2, 1989.

[112] B. Hassibi and D. Stork, "Second order derivatives for network pruning: Optimal brain surgeon," *Advances in neural information processing systems*, vol. 5, 1992.

[113] J. Martens and R. Grosse, "Optimizing neural networks with kronecker-factored approximate curvature," in *International conference on machine learning*, pp. 2408–2417, PMLR, 2015.

[114] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," *Advances in neural information processing systems*, vol. 29, 2016.

[115] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," in *Proceedings of the IEEE international conference on computer vision*, pp. 2736–2744, 2017.

[116] J. Ye, X. Lu, Z. Lin, and J. Z. Wang, "Rethinking the smaller-norm-less-informative assumption in channel pruning of convolution layers," in *International conference on learning representations*, 2018.

[117] V. Lebedev and V. Lempitsky, "Fast convnets using group-wise brain damage," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2554–2564, 2016.

[118] G. Hinton, "Distilling the knowledge in a neural network," in *Deep learning and representation learning workshop in conjunction with NIPS*, 2014.

[119] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets. rxiv preprint," *arXiv preprint arXiv:1412.6550*, 2014.

[120] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *International conference on learning representations*, 2017.

[121] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1365–1374, 2019.

[122] Z. Xiao, X. Fu, J. Huang, Z. Cheng, and Z. Xiong, "Space-time distillation for video super-resolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2113–2122, 2021.

[123] J. Lee and S. Park, "Knowledge distillation for optical flow-based video super-resolution.," *J. Comput. Sci. Eng.*, vol. 17, no. 1, pp. 13–19, 2023.

[124] Y. Hu, M. Liu, W. Yang, J. Liu, and Z. Guo, "Collaborative spatial-temporal distillation for efficient video deraining," in *2023 IEEE international conference on multimedia and expo (ICME)*, pp. 1937–1942, IEEE, 2023.

[125] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pp. 694–711, Springer, 2016.

[126] N. Zheng, Y. Xu, C.-L. Guo, C. Li, *et al.*, "Training your image restoration network better with random weight network as optimization function," *Advances in neural information processing systems*, vol. 36, pp. 1270–1282, 2023.

[127] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[128] R. K. Pandey, N. Saha, S. Karmakar, and A. Ramakrishnan, "Msce: An edge-preserving robust loss function for improving super-resolution algorithms," in *International conference on neural information processing*, pp. 566–575, Springer, 2018.

[129] S. M. Ayyoubzadeh and X. Wu, "High frequency detail accentuation in cnn image restoration," *IEEE transactions on image processing*, vol. 30, pp. 8836–8846, 2021.

[130] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: nonlinear phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[131] W.-S. Lai, J.-B. Huang, O. Wang, E. Shechtman, E. Yumer, and M.-H. Yang, "Learning blind video temporal consistency," in *European conference on computer vision*, pp. 170–185, 2018.

[132] F. Zhang, Y. Li, S. You, and Y. Fu, "Learning temporal consistency for low light video enhancement from single images," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4967–4976, 2021.

[133] S. Guo, J. Ma, X. Yang, Z. Zhang, and L. Zhang, "Toward accurate and temporally consistent video restoration from raw data," *arXiv preprint arXiv:2312.16247*, 2023.

[134] T. Wang, Y. Li, J. Peng, Y. Ma, X. Wang, F. Song, and Y. Yan, "Real-time image enhancer via learnable spatial-aware 3d lookup tables," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2471–2480, 2021.

[135] J. Liu, C.-H. Wu, Y. Wang, Q. Xu, Y. Zhou, H. Huang, C. Wang, S. Cai, Y. Ding, H. Fan, *et al.*, "Learning raw image denoising with bayer pattern unification and bayer preserving augmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 0–0, 2019.

[136] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3224–3232, 2018.

[137] F. Fuentes-Hurtado, T. Delaire, F. Levet, J.-B. Sibarita, and V. Viasnoff, "Mid3a: Microscopy image denoising meets differentiable data augmentation," in *2022 international joint conference on neural networks (IJCNN)*, pp. 1–9, IEEE, 2022.

[138] A. Abdelhamed, M. A. Brubaker, and M. S. Brown, "Noise flow: Noise modeling with conditional normalizing flows," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3165–3173, 2019.

[139] K.-C. Chang, R. Wang, H.-J. Lin, Y.-L. Liu, C.-P. Chen, Y.-L. Chang, and H.-T. Chen, "Learning camera-aware noise models," in *European Conference on Computer Vision*, pp. 343–358, Springer, 2020.

[140] J. Chen, J. Chen, H. Chao, and M. Yang, "Image blind denoising with generative adversarial network based noise modeling," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3155–3164, 2018.

[141] D.-W. Kim, J. Ryun Chung, and S.-W. Jung, "Grdn: Grouped residual dense network for real image denoising and gan-based real-world noise modeling," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 0–0, 2019.

[142] Z. Fu, L. Guo, C. Wang, Y. Wang, Z. Li, and B. Wen, "Temporal as a plugin: Unsupervised video denoising with pre-trained image denoisers," in *European conference on computer vision*, pp. 349–367, Springer, 2024.

[143] H. Yue, C. Cao, L. Liao, R. Chu, and J. Yang, "Supervised raw video denoising with a benchmark dataset on dynamic scenes," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2301–2310, 2020.

[144] N. Komodakis and S. Gidaris, "Unsupervised representation learning by predicting image rotations," in *International conference on learning representations*, 2018.

[145] G. Larsson, M. Maire, and G. Shakhnarovich, "Colorization as a proxy task for visual understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6874–6883, 2017.

[146] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

[147] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016.

[148] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *International conference on machine learning*, pp. 843–852, PMLR, 2015.

[149] D. Xu, J. Xiao, Z. Zhao, J. Shao, D. Xie, and Y. Zhuang, "Self-supervised spatiotemporal learning via video clip order prediction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10334–10343, 2019.

[150] B. Huang, J. Li, B. Yao, Z. Yang, E. Y. Lam, J. Zhang, W. Yan, and J. Qu, "Enhancing image resolution of confocal fluorescence microscopy with deep learning," *PhotoniX*, vol. 4, no. 1, p. 2, 2023.

[151] C. Qiao, Y. Zeng, Q. Meng, X. Chen, H. Chen, T. Jiang, R. Wei, J. Guo, W. Fu, H. Lu, *et al.*, "Zero-shot learning enables instant denoising and super-resolution in optical fluorescence microscopy," *Nature communications*, vol. 15, no. 1, p. 4180, 2024.

[152] H. Wang, Y. Rivenson, Y. Jin, Z. Wei, R. Gao, H. Günaydın, L. A. Bentolila, C. Kural, and A. Ozcan, "Deep learning enables cross-modality super-resolution in fluorescence microscopy," *Nature methods*, vol. 16, no. 1, pp. 103–110, 2019.

[153] Y. Wang, Z. Huang, X. Wang, F. Yang, X. Yao, T. Pan, B. Li, and J. Chu, "Real-time fluorescence imaging flow cytometry enabled by motion deblurring and deep learning algorithms," *Lab on a Chip*, vol. 23, no. 16, pp. 3615–3627, 2023.

[154] T. Tanay, A. Sootla, M. Maggioni, P. K. Dokania, P. Torr, A. Leonardis, and G. Slabaugh, "Diagnosing and preventing instabilities in recurrent video processing," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 2, pp. 1594–1605, 2022.

[155] B. N. Chiche, A. Woiselle, J. Frontera-Pons, and J.-L. Starck, "Stable long-term recurrent video super-resolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 837–846, 2022.

[156] J. Lin, C. Gan, and S. Han, "Tsm: Temporal shift module for efficient video understanding," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7083–7093, 2019.

[157] Y. Li, Z. Mi, and X. Fu, "Multiframe-to-multiframe network for underwater unpaired video enhancement," *Displays*, p. 103063, 2025.

[158] L. Xiang, J. Zhou, J. Liu, Z. Wang, H. Huang, J. Hu, J. Han, Y. Guo, and G. Ding, "Remonet: Recurrent multi-output network for efficient video denoising," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, pp. 2786–2794, 2022.

[159] V. Dewil, Z. Zheng, A. Barral, L. Raad, N. Nicolas, I. Cassagne, J.-M. Morel, G. Facciolo, B. Galerne, and P. Arias, "Adapting mimo video restoration networks to low latency constraints," in *British machine vision conference*, arXiv, 2024.

[160] K. C. Chan, S. Zhou, X. Xu, and C. C. Loy, "Basicvsr++: Improving video super-resolution with enhanced propagation and alignment," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5972–5981, 2022.

[161] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive convolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 670–679, 2017.

[162] S. Zhou, J. Zhang, J. Pan, H. Xie, W. Zuo, and J. Ren, "Spatio-temporal filter adaptive network for video deblurring," in *Proceedings of the IEEE international conference on computer vision*, 2019.

[163] B. Ji and A. Yao, "High-pass kernel prediction for efficient video deblurring," in *2025 IEEE/CVF winter conference on applications of computer vision*, pp. 2442–2452, IEEE, 2025.

[164] J.-S. Yun, M. H. Kim, H.-I. Kim, and S. B. Yoo, "Kernel adaptive memory network for blind video super-resolution," *Expert systems with applications*, vol. 238, p. 122252, 2024.

[165] B. Mildenhall, J. T. Barron, J. Chen, D. Sharlet, R. Ng, and R. Carroll, "Burst denoising with kernel prediction networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2502–2510, 2018.

[166] Z. Xia, F. Perazzi, M. Gharbi, K. Sunkavalli, and A. Chakrabarti, "Basis prediction networks for effective burst denoising with large kernels," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11844–11853, 2020.

[167] W. Cho, S. Son, and D.-S. Kim, "Weighted multi-kernel prediction network for burst image super-resolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 404–413, 2021.

[168] J. Cao, Y. Li, K. Zhang, and L. Van Gool, "Video super-resolution transformer," *arXiv preprint arXiv:2106.06847*, 2021.

[169] 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.

[170] E. Lewandowska, A. Jezierska, and D. Węsierski, "Streamed optical flow adaptation from synthetic to real dental scenes," *IEEE transactions on medical imaging*, 2025.

[171] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8008–8017, 2020.

[172] E. Katsaros, P. K. Ostrowski, K. Włódarczak, E. Lewandowska, J. Ruminski, D. Siupka-Mróz, Ł. Lassmann, A. Jezierska, and D. Węsierski, "Multitask video enhancement for dental interventions," in *International conference on medical image computing and computer-assisted intervention*, pp. 177–187, Springer, 2022.

[173] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable convnets v2: More deformable, better results," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9308–9316, 2019.

[174] A. Davy, T. Ehret, J.-M. Morel, P. Arias, and G. Facciolo, "A non-local cnn for video denoising," in *2019 IEEE international conference on image processing (ICIP)*, pp. 2409–2413, IEEE, 2019.

[175] G. Vaksman, M. Elad, and P. Milanfar, "Patch craft: Video denoising by deep modeling and patch matching," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2157–2166, 2021.

[176] R. Franzen, "Kodak lossless true color image suite." `http://r0k.us/graphics/kodak/`. Accessed: 2025-10-08.

[177] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Flickr2k dataset," 2024.

[178] 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.

[179] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1775–1787, 2023.

[180] T. Plotz and S. Roth, "Benchmarking denoising algorithms with real photographs," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1586–1595, 2017.

[181] A. Abdelhamed, S. Lin, and M. S. Brown, "A high-quality denoising dataset for smartphone cameras," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1692–1700, 2018.

[182] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1683–1691, 2016.

[183] J. Xu, H. Li, Z. Liang, D. Zhang, and L. Zhang, "Real-world noisy image denoising: A new benchmark," *arXiv preprint arXiv:1804.02603*, 2018.

[184] J. Anaya and A. Barbu, "Renoir–a dataset for real low-light image noise reduction," *Journal of visual communication and image representation*, vol. 51, pp. 144–154, 2018.

[185] C. Wei, W. Wang, W. Yang, and J. Liu, "Deep retinex decomposition for low-light enhancement," in *British machine vision conference (BMVC)*, 2018.

[186] C. Chen, Q. Chen, J. Xu, and V. Koltun, "Learning to see in the dark," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3291–3300, 2018.

[187] J. Cai, S. Gu, and L. Zhang, "Learning a deep single image contrast enhancer from multi-exposure images," *IEEE transactions on image processing*, vol. 27, no. 4, pp. 2049–2062, 2018.

[188] S. Roth and M. J. Black, "Fields of experts: A framework for learning image priors," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 2, pp. 860–867, IEEE, 2005.

[189] H. R. Künsch, "Robust priors for smoothing and image restoration," *Annals of the Institute of Statistical Mathematics*, vol. 46, no. 1, pp. 1–19, 1994.

[190] S. G. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE transactions on image processing*, vol. 9, no. 9, pp. 1532–1546, 2000.

[191] L. Kaur, S. Gupta, and R. C. Chauhan, "Image denoising using wavelet thresholding.," in *ICVGIP*, vol. 2, pp. 16–18, 2002.

[192] R. Oktem and N. Ponomarenko, "Image filtering based on discrete cosine transform," *Telecommunications and radio engineering*, vol. 66, no. 18, 2007.

[193] V. V. Lukin, D. V. Fevralev, N. N. Ponomarenko, S. K. Abramov, O. Pogrebnyak, K. O. Egiazarian, and J. T. Astola, "Discrete cosine transform–based local adaptive filtering of images corrupted by nonstationary noise," *Journal of Electronic Imaging*, vol. 19, no. 2, pp. 023007–023007, 2010.

[194] J.-L. Starck, E. J. Candès, and D. L. Donoho, "The curvelet transform for image denoising," *IEEE transactions on image processing*, vol. 11, no. 6, pp. 670–684, 2002.

[195] W. Aili, Z. Ye, M. Shaoliang, and Y. Mingji, "Image denoising method based on curvelet transform," in *2008 3rd IEEE conference on industrial electronics and applications*, pp. 571–574, IEEE, 2008.

[196] R. Eslami and H. Radha, "Translation-invariant contourlet transform and its application to image denoising," *IEEE transactions on image processing*, vol. 15, no. 11, pp. 3362–3374, 2006.

[197] R. Eslami and H. Radha, "The contourlet transform for image denoising using cycle spinning," in *The Thrity-Seventh asilomar conference on signals, systems & computers, 2003*, vol. 2, pp. 1982–1986, IEEE, 2003.

[198] A. P. Witkin, "Scale-space filtering," in *Readings in computer vision*, pp. 329–332, Elsevier, 1987.

[199] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, pp. 839–846, IEEE, 1998.

[200] T. Huang, G. Yang, and G. Tang, "A fast two-dimensional median filtering algorithm," *IEEE transactions on acoustics, speech, and signal processing*, vol. 27, no. 1, pp. 13–18, 1979.

[201] J. Jiang and J. Shen, "An effective adaptive median filter algorithm for removing salt & pepper noise in images," in *2010 Symposium on photonics and optoelectronics*, pp. 1–4, IEEE, 2010.

[202] K. K. V. Toh and N. A. M. Isa, "Noise adaptive fuzzy switching median filter for salt-and-pepper noise reduction," *IEEE signal processing letters*, vol. 17, no. 3, pp. 281–284, 2009.

[203] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 2, pp. 60–65, Ieee, 2005.

[204] 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.

[205] V. Jain and S. Seung, "Natural image denoising with convolutional networks," *Advances in neural information processing systems*, vol. 21, 2008.

[206] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with bm3d?," in *2012 IEEE conference on computer vision and pattern recognition*, pp. 2392–2399, IEEE, 2012.

[207] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[208] 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.

[209] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, "Toward convolutional blind denoising of real photographs," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1712–1722, 2019.

[210] T. Plötz and S. Roth, "Neural nearest neighbors networks," *Advances in neural information processing systems*, vol. 31, 2018.

[211] S. Bako, T. Vogels, B. McWilliams, M. Meyer, J. Novák, A. Harvill, P. Sen, T. Derose, and F. Rousselle, "Kernel-predicting convolutional networks for denoising monte carlo renderings.," *ACM transactions on graphics*, vol. 36, no. 4, pp. 97–1, 2017.

[212] R. Ma, S. Li, B. Zhang, and Z. Li, "Generative adaptive convolutions for real-world noisy image denoising," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, pp. 1935–1943, 2022.

[213] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, "Restormer: Efficient transformer for high-resolution image restoration," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5728–5739, 2022.

[214] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "Swinir: Image restoration using swin transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1833–1844, 2021.

[215] Y. Quan, M. Chen, T. Pang, and H. Ji, "Self2self with dropout: Learning self-supervised denoising from single image," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1890–1898, 2020.

[216] T. Huang, S. Li, X. Jia, H. Lu, and J. Liu, "Neighbor2neighbor: Self-supervised denoising from single noisy images," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14781–14790, 2021.

[217] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, "The 2017 DAVIS challenge on video object segmentation," *arXiv:1704.00675*, 2017.

[218] M. Tassano, J. Delon, and T. Veit, "DVDnet: A fast network for deep video denoising," in *2019 IEEE international conference on image processing*, pp. 1805–1809, IEEE, 2019.

[219] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *arXiv preprint arXiv:1603.00831*, 2016.

[220] H. Jiang and Y. Zheng, "Learning to see moving objects in the dark," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7324–7333, 2019.

[221] R. Wang, X. Xu, C.-W. Fu, J. Lu, B. Yu, and J. Jia, "Seeing dynamic scene in the dark: A high-quality video dataset with mechatronic alignment," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9700–9709, 2021.

[222] Y. Fu, Z. Wang, T. Zhang, and J. Zhang, "Low-light raw video denoising with a high-quality realistic motion dataset," *IEEE transactions on multimedia*, vol. 25, pp. 8119–8131, 2022.

[223] H. Yue, C. Cao, L. Liao, and J. Yang, "Rvideformer: Efficient raw video denoising transformer with a larger benchmark dataset," *IEEE transactions on circuits and systems for video technology*, 2025.

[224] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, "Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms," *IEEE transactions on image processing*, vol. 21, no. 9, pp. 3952–3966, 2012.

[225] T.-W. Chan, O. C. Au, T.-S. Chong, and W.-S. Chau, "A novel content-adaptive video denoising filter," in *Proceedings.(ICASSP'05). IEEE international conference on acoustics, speech, and signal processing, 2005.*, vol. 2, pp. ii–649, IEEE, 2005.

[226] S. Reeja and N. Kavya, "Real time video denoising," in *2012 IEEE international conference on engineering education: innovative practices and future trends (AICERA)*, pp. 1–5, IEEE, 2012.

[227] D. Rusanovskyy and K. Egiazarian, "Video denoising algorithm in sliding 3d dct domain," in *International conference on advanced concepts for intelligent vision systems*, pp. 618–625, Springer, 2005.

[228] F. Luisier, T. Blu, and M. Unser, "Sure-let for orthonormal wavelet-domain video denoising," *IEEE transactions on Circuits and Systems for Video Technology*, vol. 20, no. 6, pp. 913–919, 2010.

[229] X. Chen, L. Song, and X. Yang, "Deep rnns for video denoising," in *Applications of digital image processing XXXIX*, vol. 9971, pp. 573–582, SPIE, 2016.

[230] M. Claus and J. Van Gemert, "Videnn: Deep blind video denoising," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 0–0, 2019.

[231] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "Deepflow: Large displacement optical flow with deep matching," in *Proceedings of the IEEE international conference on computer vision*, pp. 1385–1392, 2013.

[232] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *European conference on computer vision*, pp. 611–625, Springer, 2012.

[233] L. Lindner, A. Effland, F. Ilic, T. Pock, and E. Kobler, "Lightweight video denoising using aggregated shifted window attention," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 351–360, 2023.

[234] X. Wang, K. C. Chan, K. Yu, C. Dong, and C. Change Loy, "Edvr: Video restoration with enhanced deformable convolutional networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 0–0, 2019.

[235] C. Chen, Q. Chen, M. N. Do, and V. Koltun, "Seeing motion in the dark," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3185–3194, 2019.

[236] J. Li, X. Wu, Z. Niu, and W. Zuo, "Unidirectional video denoising by mimicking backward recurrent modules with look-ahead forward ones," in *European Conference on Computer Vision*, pp. 592–609, Springer, 2022.

[237] P. K. Ostrowski, E. Katsaros, D. Węsierski, and A. Jezierska, "BP-EVD: Forward block-output propagation for efficient video denoising," *IEEE transactions on image processing*, vol. 31, pp. 3809–3824, 2022.

[238] 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.

[239] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

[240] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1759–1768, 2021.

[241] L. Chen, X. Chu, X. Zhang, and J. Sun, "Simple baselines for image restoration," in *European conference on computer vision*, pp. 17–33, Springer, 2022.

[242] H. Shen, Z.-Q. Zhao, and W. Zhang, "Adaptive dynamic filtering network for image denoising," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, pp. 2227–2235, 2023.

[243] A. Ghasemabadi, M. K. Janjua, M. Salameh, C. ZHOU, F. Sun, and D. Niu, "Cascadedgaze: Efficiency in global context extraction for image restoration," *Transactions on machine learning research*, 2024.

[244] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," in *European conference on computer vision*, pp. 402–419, Springer, 2020.

[245] H. Shen, Z.-Q. Zhao, and W. Zhang, "Adaptive dynamic filtering network for image denoising," in *Proceedings of the AAAI conference on artificial intelligence*, 2023.

[246] P. K. Ostrowski, D. Węsierski, A. Jezierska, and T. Stefański, "Lifting deep image denoisers to video with frame interpolation pre-training," *IEEE transactions on circuits and systems for video technology*, 2025.

[247] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4161–4170, 2017.