POLITECNICO DI TORINO
Repository ISTITUZIONALE

A multiplication-free neural architecture for image restoration

Original
A multiplication-free neural architecture for image restoration / Dordoni, L., Valsesia, D., Magli, E.. - In: NEUROCOMPUTING. - ISSN 0925-2312. - 684:(2026). [10.1016/j.neucom.2026.133579]

Availability:
This version is available at: 11583/3009910 since: 2026-04-15T13:28:30Z

Publisher:
Elsevier

Published
DOI:10.1016/j.neucom.2026.133579

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

27 November 2026

Neurocomputing 684 (2026) 133579

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A multiplication-free neural architecture for image restoration

Luca Dordoni, Diego Valsesia*, Enrico Magli

Department of Electronics and Telecommunications, Politecnico di Torino, Corso Duca degli Abruzzi, 24, Turin, 10129, Italy

HIGHLIGHTS

- Fully multiplication-free neural network backbone for image restoration.
- Novel LayerNorm avoiding multiplications via average absolute deviation.
- Annealed training progressively adapts model from full-precision to quantized.
- Combinations of bitshift operations fully replace multiplications.
- Achieves competitive performance using only addition and bitwise operations.

ARTICLE INFO

Communicated by G. Meng

Keywords:
Quantization
Multiplication-free
Image restoration
Deblurring
Denoising

ABSTRACT

Deep neural networks have established themselves as the dominant approach for image restoration problems such as deblurring and denoising. However, the high computational complexity of state-of-the-art designs prevents their effective use in resource-constrained scenarios such as edge devices, where power-efficient inference is key. In this paper, we present a state-of-the-art backbone neural network design for image restoration, called MuFIR (Multiplication-Free Image Restoration), that is entirely devoid of multiplication operations. When coupled with suitable hardware implementations, the proposed concept enables fast and low-complexity inference by requiring only integer additions and bit shifts. This is made possible by several ingredients proposed in this work, namely ternary weight quantization to eliminate multiplications in the main network layers, careful use of novel normalizations to ensure stability of the ternarized architecture, and quantization of specific parameters and activations to combinations of powers of two, to remove the remaining multiplications. This is coupled with an annealed training procedure which progressively transforms a conventional network into our multiplication-free design. We experimentally show that, despite the all-integer operations and the lack of multiplications, MuFIR achieves performance close to that of full-precision models in terms of deblurring and denoising quality.

1. Introduction

Image restoration [1–9] tasks such as deblurring, denoising, super-resolution, etc. are inverse problems in which an image has to be reconstructed from partial or degraded observations. Deep neural networks have been highly successful at such low-level vision tasks thanks to learning sophisticated image priors from data. Indeed, such models [1–3,6,9] exploit operations like convolution to model locality of features, as well as attention operations [7,8] that can capture non-local self-similar patterns. However, the complexity of state-of-the-art image restoration networks has steadily grown over the last few years, posing challenges for their execution in resource-constrained environments like edge devices.

Numerous methods have emerged in the neural networks literature in response to complexity challenges, some of which have also been adopted in image restoration models. An established approach is to reduce the numerical precision of operations performed by neural network layers, e.g., via quantization of their weights and activations. In particular, Quantization-Aware Training (QAT) allows a model to adapt its parameters to the quantized regime during training, bypassing the non-differentiable discretization operations with approximations like straight-through estimators (STE) [10]. This typically results in superior performance at low bit-widths compared to simple Post-Training Quantization [11–17]. Quantization can be brought to its extreme setting of binarization, where weights and possibly activations assume just two values [11–13,18,19]. While this allows designing neural networks

* Corresponding author.
Email address: diego.valsesia@polito.it (D. Valsesia).

https://doi.org/10.1016/j.neucom.2026.133579
Received 1 October 2025; Received in revised form 19 March 2026; Accepted 4 April 2026
Available online 6 April 2026
0925-2312/© 2026 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

that only employ simple logic gates, it often incurs significant penalties in terms of model quality, especially in image restoration [20–22] where models are typically more sensitive to the reduction of numerical precision.

Beyond extreme binarization, there has been growing interest in alternative strategies for hardware efficiency, particularly in minimizing the number of multiplications performed by the model, as these represent a primary bottleneck. The main ingredient of this approach is the use of ternary weights, i.e., weights belonging to the $\{-1, 0, 1\}$ set, as they allow implementing a layer operation with only signed additions. Indeed, older works on image classification [15,16,23], where ternarization has been applied to convolutions, and recent works on LLMs [14], where ternarization has been applied to linear operators, show that this approach can be extremely effective. Consequently, a recent line of research is shaping around developing architectures designed to significantly reduce or even eliminate multiplication operations [24–26]. However, although these methods considerably reduce the use of multiplications, they frequently do not eliminate them entirely. Many works rely on attention modules and normalization layers that make critical use of multiplications and more complex operations such as square-root and exponentiation.

The exploration of architectures with reduced multiplications in the image restoration field is currently very limited, with the only notable example being AdderSR [27], which applies AdderNet [25], a method that replaces standard convolutions with element-wise additions based on the L1 distance between inputs and weights, to the task of super-resolution.

At the same time, growing evidence suggests that ternary, multiplication-free neural networks combined with dedicated designs for hardware accelerators can unlock significant improvements in efficiency. In particular Scherer et al. [28] propose a fully digital ASIC for ternary neural networks which reports 3.1 POp/s/W energy efficiency, with reduced inference cost by a factor of 4.8% up to 21%, specifically highlighting the benefits of using ternary neural networks with respect to binary networks as they allow for sparse weights that reduce switching activity. Moreover, Rutishauser et al. [29] point out that ternary inference yields better accuracy-energy trade-off in comparison to their binary counterpart, and they introduce a lightweight RISC-V ISA extension for ternary convolutions and prove that specialized ternary instructions can yield 67% higher throughput than optimized 2-bit integer kernels, while increasing only 5.2%, which corresponds to 57.1% improvement in energy efficiency.

In this paper, we introduce MuFIR, a backbone neural network for image restoration. Unlike many existing methods that still rely on multiplications for operations like normalization and attention, MuFIR is designed to be entirely multiplication-free across all its layers. To the best of our knowledge, this is the first attempt to entirely eliminate multiplications from an image restoration model. While in this paper we propose the theoretical framework in terms of neural architecture and operations design to achieve a multiplication-free model, we remark that its gains in terms of inference speed or power consumption can only be observed with dedicated hardware designs such as FPGAs or ASICs, rather than with conventional processing kernels on existing CPUs/GPUs. Our key contributions towards a multiplication-free image restoration neural network can be summarized as follows:

- Ternary quantization: we extend the trend of low-precision networks by applying weight ternarization to architectures specifically tailored for image restoration tasks.
- Normalization: we carefully expand the use of normalization operations to stabilize the ternary network and provide a new definition for LayerNorm which does not use multiplications (MuFLN).
- Multiplication-free inference: all remaining multiplications are eliminated via quantization to combinations of bitshift operations (APO2 quantization).
- Multiplication-free annealed training: an annealed training methodology allows the model to progressively adapt to its fully multiplication-free operational constraints.

2. Background

Image restoration tasks are a form of inverse problems, in which a neural network is used to map a degraded image back to its high-quality counterpart. The rapid advancement of deep learning has significantly impacted the development of models for image restoration and related tasks, with early approaches leveraging convolutional neural networks (CNNs) achieving notable success [1–3]. More recently, architectural complexity has increased, incorporating mechanisms such as self-attention [30], and the emerging state-space-based Mamba [31]. These advancements have given rise to a new generation of complex image restoration architectures, such as SwinIR [7], Restormer [8], MPRNet [6], MIMO-UNet [32] and NAFNet [9]. Recently, Mamba has been integrated into image restoration architectures [33], as well as diffusion-guided restoration models [34,35]. These increasingly complex architectures have been employed across a range of specialized tasks, including image deblurring and denoising [5,36–39].

In parallel to this trend, another line of research aims at designing efficient networks that minimize both computational cost and memory footprint. Quantization has emerged as a key strategy in this domain, with binary neural networks (BNNs) representing its most extreme form by constraining weights and activations to $\{-1, +1\}$ [11,12,18]. Pioneering work in the field of image restoration includes the application of binary convolutions [20] for single-image super-resolution, using a ResNet backbone [40]. Xin et al. [21] introduced a bit-accumulation scheme to mitigate accuracy degradation, while [41] enhanced the low-precision compatibility by removing all batch-normalization layers. More recently, Xia et al. [22] conducted a systematic study of BNNs for image restoration tasks, deconstructing the standard binary block and improving performance for binary single-image super-resolution and binary image denoising. Despite their success at reducing computational load through binarization, BNNs are not completely multiplication-free. Although [22,41] come close to eliminating multiplications, they still rely on various scaling factors in their binarization processes, with [22] also introducing an RPReLU activation defined by its own scale parameters. Likewise, Ma et al. [20] and Xin et al. [21] continue to incorporate multiplication-dependent operations in their batch normalization and binary scaling parameters. Current binary neural networks for image restoration also do not leverage recent advancements in neural architectures, and the binarization process typically significantly degrades the quality of the outputs.

Concurrently, the literature on large language models has seen substantial efforts to optimize transformer architectures with the goal of multiplication-free inference, as evidenced by recent works on piecewise affine operations [42], MatMul-free LM [26], and ShiftAddLLM [43].

3. Method

This section introduces MuFIR, our proposed multiplication-free neural network backbone for image restoration. An overview is shown in Fig. 1. We remark that our design builds upon the NAFNet architecture [9], which we adapt to be entirely multiplication-free. This choice is motivated by the fact that NAFNet provides an already simple, yet state-of-the-art architecture, and, particularly, it replaces the Transformer's self-attention operations and activation functions with simple gating and simplified attention mechanisms. Its minimal reliance on complex operations makes it more suitable to be readily transformed into a fully multiplication-free architecture. We also remark that we propose a

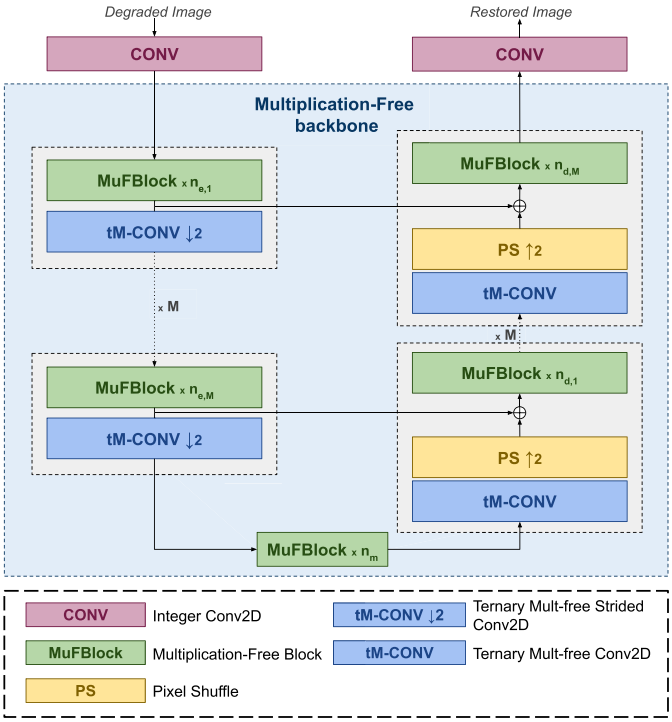


Fig. 1. MuFIR architecture. First and last convolutions are regular integer-quantized. The multiplication-free backbone follows a U-Net layout relying on the repetition of MuFBlocks. Feature maps are downsampled by ternary multiplication-free strided convolution and upsampled by pixel shuffling. This downsampling/upsampling multiresolution process is repeated $M = 4$ times.

backbone that is multiplication-free, meaning that the first and last convolutional layers are classic layers, albeit integer-quantized. This is a common setting throughout the literature on binarization, as these layers are needed to properly process the input and generate the output. Furthermore, for clarity of presentation, we assume all operations are performed with integer arithmetic with arbitrary precision, i.e. accumulators/adders as wide as necessary. In practice, one could bound their precision with further requantization operations [44].

In the following, we first introduce the architecture and how it differs from NAFNet in its main blocks. Then, we discuss the ingredients used by the various layers to make it multiplication-free, namely weight ternarization, multiplication-free LayerNorm (MuFLN), APO2 quantization and multiplication-free annealed training.

3.1. Overall architecture

We refer the reader to [9] for details on the NAFNet architecture, which serves as foundation for our proposed MuFIR architecture. As shown in Fig. 1, at a high level, the design follows NAFNet's U-Net paradigm with M multiresolution stages which progressively downsample and upsample the input image, with additive skip connections at each resolution. Each resolution is processed by the repetition of multiplication-free blocks (MuFBlocks) depicted in Fig. 2. Overall, a MuFBlock closely follows a NAFNet block with some important differences. First, extra LayerNorm (LN) [45] operations are coupled with every convolution operation. This is needed to improve network stability since convolutions will be implemented with ternary weights as detailed in Section 3.2. Moreover, conventional LN is replaced with the MuFLN detailed in Section 3.3. Notice that the SimpleGate operation splits the feature channels into two sets and multiplies them together. Similarly, the channel attention operation has a multiplicative nature. In these cases, we use the APO2 quantization mechanism explained in Section 3.4 to avoid multiplications. Finally, the average pooling operations require input feature maps to have spatial dimensions that are powers of two, which avoids the need for explicit division.

3.2. Weight ternarization

Convolutions account for the bulk of operations performed by our network. We propose to ternarize the weights[1] of such convolutional layers in order to implement the operations with integer adders and accumulators.

Let $\mathbf{W}$ be a floating-point full-precision weight tensor, then we define the scaling factor s as:

$$s = \frac{1}{N}\sum_{i=1}^{C_{\text{out}}}\sum_{j=1}^{C_{\text{in}}}\sum_{m=1}^{K_h}\sum_{n=1}^{K_w}\left|W_{i,j,m,n}\right| \tag{1}$$

where $N = C_{\text{out}} \cdot C_{\text{in}} \cdot K_h \cdot K_w$ is the number of elements in the weight tensor. Ternary Quantization is then achieved with the following set of operations:

$$\bar{\mathbf{W}} = \frac{\mathbf{W}}{s}, \tag{2}$$

$$\tilde{\mathbf{W}} = \text{clip}\left(\text{round}(\bar{\mathbf{W}}), -1, 1\right). \tag{3}$$

Here, $\tilde{\mathbf{W}} \in \{-1, 0, +1\}^N$ is the ternary tensor obtained by rounding and clipping. This is precomputed at the end of training and stored to be used in inference, together with the scaling factor s. Notice that, conceptually, we need to multiply the output of the convolution operation by the scaling factor s in order to restore the output scale seen during training. For this reason, the scaling factor s is APO2-quantized according to the procedure in Section 3.4 to replace its product with combination of bitshifts, avoiding multiplications.

As will be shown in Section 4.3, direct weight ternarization of the NAFNet architecture leads to training instability and suboptimal results. This instability arises because the single scaling factor s is insufficient to effectively normalize activations and gradients, especially when multiple layers are stacked between successive LayerNorm operations. For this reason, we always couple a multiplication-free LayerNorm operation with a ternary convolutional layer, as shown in Fig. 1.

3.3. Multiplication-free LayerNorm (MuFLN)

It is common practice in the literature [20,21,26,46], even in works striving to reduce multiplications, to leave normalization layers to their floating-point mathematical definitions which include multiplications, divisions, and powers. This is due to a perceived sensitivity of these operations to quantization. We introduce a novel definition for the LayerNorm layer which is multiplication (and other complex operations)-free and shows strong experimental performance. This definition relies on the computation of average absolute deviations (AAD) instead of standard deviations to avoid squaring and APO2-quantization to avoid multiplications in normalization.

Let us denote $\mathbf{x} \in \mathbb{R}^{N\times C\times H\times W}$ as the input tensor to the MuFLN layer, then we compute

$$\mu_{n,h,w} = \frac{1}{C}\sum_{c=1}^{C} x_{n,c,h,w}, \tag{4}$$

$$\text{AAD}_{n,h,w} = \text{APO2}\left(\frac{1}{C}\sum_{c=1}^{C}\left|x_{n,c,h,w} - \mu_{n,h,w}\right|\right), \tag{5}$$

$$\rho_{n,h,w} = \text{APO2}\left(\frac{1}{\text{AAD}_{n,h,w}}\right), \tag{6}$$

$$y_{n,c,h,w} = \gamma_c\left(x_{n,c,h,w} - \mu_{n,h,w}\right)\rho_{n,h,w} + \beta_c. \tag{7}$$

where γ_c, β_c are per-channel learnable parameters. γ_c entries are initialized to $\frac{1}{\sqrt{2}}$ for training and the final values are APO2-quantized, while

[1] Biases are integer-quantized as they are not involved in multiplications.

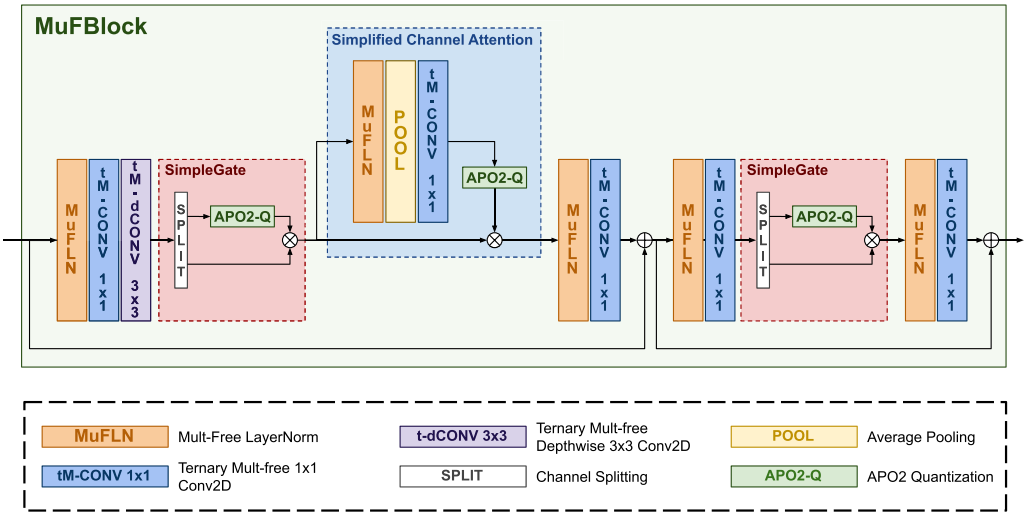


Fig. 2. The MuFBlock. It contains multiple MuFLN layers for multiplication-free normalization, ternary multiplication-free convolutions, APO2-quantized gating operations, APO2-quantized simplified channel attention, and residual connections.

final values for β_c are just integer-quantized. Note that C is assumed to be a power of 2.

3.4. APO2 quantization

While ternary weights and the use of AAD as normalization statistics can substantially reduce the number of multiplications, there are several other operations in the neural network that require a large number of multiplications. We have already seen how ternary convolutions apply a rescaling operation and normalizations perform division by the AAD statistic. Moreover, operations such as the SimpleGate and channel attention also theoretically employ multiplications.

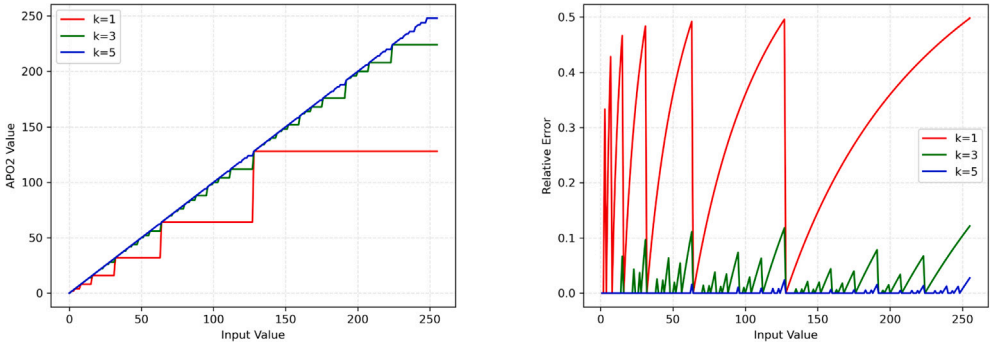
In order to achieve a truly multiplication-free backbone, we propose defining a representation of integers as a combination of a small number of powers of 2 (Approximation by Powers of 2 - APO2). To clarify the reasoning, let us consider the product of two integers a and b where the latter can be written as a sum of powers of 2; then the product can be computed via bit shifts and additions as

$$ab = a\left(\sum_i x_i 2^i\right) = \sum_i x_i (a \ll i) \qquad x_i \in \{0,1\} \tag{8}$$

In the proposed network, we can distinguish between values requiring APO2 quantization that are fixed from training and those that are computed at runtime. Examples of the former are the scaling factor s for ternary convolution and parameters γ_c in MuFLN. The latter case instead accounts for one branch of the SimpleGate layer, the attention branch in channel attention, and the reciprocal AAD value in MuFLN. For pretrained values, when their floating-point representation obtained by training is smaller than one, we consider using combinations of right instead of left bitshifts. All values computed at runtime bitshift to the left due to input scaling factors that ensure sufficient precision.

In practice, APO2 Quantization can be implemented by lookup tables, where for every integer in a certain dynamic range the best approximation using k powers of two is stored. Hyperparameter k controls the coarseness of this quantization, and it is up to the system designer. The dynamic range covered by the lookup table is also a design hyperparameter, which controls the coarseness of the operation. In fact, if values exceed the predetermined range, they can be requantized to the target range (by power-of-2 rescaling), with potentially increased error. In our experiments, we always used a dynamic range of $[0, 255]$ and have observed that $k = 3$ is sufficient to achieve good results, while requiring only 3 bit shifts.

Fig. 3 visually depicts the quantizer levels and the relative error achieved for various values of hyperparameter k. The total memory required for the lookup tables is on the order of a few hundred bytes, as detailed in Table 1. This is obtained by considering that the input is an 8-bit integer which is mapped into k bit shifts, i.e., k values between 0 and 7, each representable using 3 bits. We also remark that



(a) APO2 approximation. (b) Relative approximation errors.

Fig. 3. Quantization levels and relative approximation errors for hyperparameter $k \in (1, 3, 5)$.

Table 1
Lookup table memory requirements for $k \in (1, 3, 5)$.

Hyperparameter k	Memory
$k = 1$	96 bytes
$k = 3$	288 bytes
$k = 5$	480 bytes

its implementation is a straightforward O(1) memory lookup where the magnitude of the activation value is used as key to the corresponding memory location, so its complexity is indeed negligible.

3.5. Multiplication-free annealed training

As previously discussed, MuFIR achieves its multiplication-free design through extensive quantization. This involves not only the ternarization of weights but also the quantization of some activations and other parameters to powers of two (APO2). Therefore, it is critical that the training process is explicitly aware of these constraints to ensure effective model optimization. For this reason, we adopt a QAT strategy, where both ternarization and APO2 quantization use STEs for gradient propagation.

However, the very coarse quantization performed throughout the network makes the classic QAT procedure fail. For this reason, we adopt an annealed training procedure that blends the contributions of the full-precision network and the quantized network, while making the latter more relevant over time. In practice, for each training iteration, the forward pass employs values interpolated between the full-precision and the fully-quantized ones. This process is controlled by an interpolation factor α:

$$\theta(\alpha) = (1 - \alpha)\,\theta^{\mathrm{FP}} + \alpha\,\theta^{\mathrm{Q}}, \quad \alpha \in [0, 1], \tag{9}$$

where θ^{Q} represents the quantized (ternary or APO2) parameter/activation and θ^{FP} is its full-precision version. The interpolation factor α is initialized to 0 at the beginning of training and then linearly increased until it reaches the value of 1 at approximately 75% of the training duration. This allows the model to use a better initialization and then optimize itself for the extensively quantized regime. Throughout the training process, STEs are used for optimization.

4. Experimental results

In this section, we present experimental results to evaluate the effectiveness of MuFIR and validate our design choices. We follow standard practice in image restoration literature by evaluating MuFIR on image deblurring (GoPro dataset [51]) and RGB image denoising (SIDD [52] dataset) benchmarks. We employ a NAFNet backbone whose internal block expansion factor is equal to 64. The network contains a total of 36 MuFBlocks and 4 multiresolution stages. Using the notation presented in Fig. 1 these are allocated as $n_e = [n_{e,1}, n_{e,2}, n_{e,3}, n_{e,4}]$ in the encoding path, n_m in the bottleneck, and $n_d = [n_{d,1}, n_{d,2}, n_{d,3}, n_{d,4}]$ in the decoding path. The training protocol follows [9]: we train the network with pairs of degraded and ground-truth patches of 256×256 pixels, in batches of 32, with flipping, rotation, and random cropping augmentations. We apply gradient clipping and optimize a PSNR-based loss. We train the network using an AdamW optimizer, ensuring stable convergence under our highly quantized regime. The initial learning rate is equal to 10^{-3}, weight decay is set to 10^{-3} with momentum coefficients $\beta_1 = \beta_2 = 0.9$. Training uses a multistep learning rate schedule specifically designed to aid the annealed training scheme: the learning rate decays at predetermined milestones in parallel with the linear growth of the interpolation parameter α, improving weight updates and preventing large fluctuations. The optimization proceeds for 260K iterations, reducing the learning rate to 10^{-7} with scheduled drops at iterations [50K, 120K, 170K, 220K], for two steps of gradient accumulation. We base the evaluation of the model on two commonly used metrics in image restoration, Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM). Our architecture is trained on 4 NVIDIA A40 GPUs for approximately 9 days.

4.1. Image deblurring

We evaluate our deblurring model on the GoPro dataset [51], which consists of pairs of sharp (ground-truth) and blurred images. The blurred frames are created by averaging consecutive frames of a dynamic scene captured with a high-speed camera. MuFBlocks are distributed across the network as $n_e = [1, 1, 1, 28]$, $n_m = 1$, and $n_d = [1, 1, 1, 1]$. Our MuFIR is the first multiplication-free architecture for image restoration, hence, there exist no direct one-to-one baselines for comparisons. In order to provide perspective on its performance, we decided to compare it to existing literature on state-of-the-art binary architectures for image restoration. We remark that weight binarization leads to severe performance degradation. Moreover, as discussed in Section 2 these architectures are not completely multiplication-free. The examined schemes include BBCU [22], ReActNet [12], Bi-Real [13], BN-Free BNN [53] and IRNet [19]. Following [22], we test such binary methods on a DnCNN [3] backbone, quantizing its body and leaving the head and the reconstruction modules in full-precision. We use the official code released by the authors for each of the methods to train them on the GoPro data. We remark that the binary methods we include as baselines are tightly coupled to the specific backbones they were originally designed around. They rely on ad-hoc architectural block changes, hence they do not transfer in a plug-and-play way to more modern/complex architectures such as NAFNet. To address backbone fairness we instead provide a naive-ternary NAFNet baseline as reported in the ablations in Table 6, using same backbone and training setup as MuFIR but without APO2 quantization. Additionally, we report two interesting points of comparison, namely the original NAFNet architecture in FP32 precision and an 8-bit integer quantized version of it. Notice that the latter still requires multiplications. These comparisons allow us to gain perspective on how much the multiplication-free constraints degrade the theoretically-achievable performance of the model at the basis of MuFIR.

We report results in Table 2. MuFIR significantly outperforms the other binary architectures, achieving superior outcomes while being completely multiplication-free. We noticed that some binary architectures struggle to perform well on the deblurring task. It should be remarked that most of these architectures were originally presented for the super-resolution problem, so they might not generalize well to deblurring. We can also notice that compared to binary architectures, MuFIR is not too far from the performance of the original FP32 NAFNet with a PSNR degradation of only 2.36 dB. To further put this into perspective, Table 3 shows how MuFIR compares with respect to slightly older FP32 state-of-the-art models, showing even smaller gaps, and, impressively, MuFIR outperforming state-of-the-art FP32 architectures from just a few years ago.

Table 2
Quantitative comparison on the GoPro dataset for image deblurring. Results are reported in terms of PSNR (dB) and SSIM. The best results are highlighted in bold.

Method	FP32-NAFNet	INT8-NAFNet	BBCU	ReActNet	Bi-Real	BN-Free	IRNet	MuFIR
PSNR (dB) ↑	33.69	33.26	26.39	25.62	24.57	25.64	26.22	**31.33**
SSIM ↑	0.967	0.964	0.807	0.790	0.778	0.790	0.802	**0.947**

Table 3
Quantitative comparison on the GoPro dataset with recent state-of-the-art FP32 architectures.

Method	PSNR	ΔPSNR	SSIM	ΔSSIM
MuFIR (ours - mult-free)	31.33 dB	–	0.947	–
NAFNet (FP32) [9]	33.69 dB	+2.36 dB	0.967	+0.020
Restormer (FP32) [8]	32.92 dB	+1.59 dB	0.961	+0.014
Mimo-Unet+ (FP32) [32]	32.45 dB	+1.12 dB	0.957	+0.010
SPAIR (FP32) [47]	32.06 dB	+0.73 dB	0.953	+0.006
Suin et al. (FP32) [48]	31.85 dB	+0.52 dB	0.948	+0.001
MT-RNN (FP32) [49]	31.15 dB	−0.18 dB	0.945	−0.002
Gao et al. (FP32) [50]	30.90 dB	−0.43 dB	0.935	−0.012

Moreover, Table 4 reports the number of multiplications, additions and bitshifts performed by the compared architectures to more explicitly contextualize the complexity of MuFIR. In reading the table, it is worth keeping in mind that multipliers are slower and more complex than adders or binary operations, that floating point operations require more complex circuitry than integer operations and that energy consumption and speed ultimately depend on the specific hardware design. We remark that it is not possible to fairly measure speed or energy efficiency with existing CPU or GPU accelerators, as their hardware is not designed for multiplication-free operations. We refer the reader to [28,29] for a study of the efficiency gains provided by multiplication-free designs on ad-hoc hardware accelerators.

A few qualitative results are displayed in Fig. 4.

4.2. Real world image denoising

We also test MuFIR on the SIDD dataset [52], which comprises paired noisy and clean images for real-world denoising. In accordance with [9], we arrange $n_e = [2, 2, 4, 8]$ MuFBlocks for the encoding path, $n_m = 12$ in the bottleneck and $n_d = [2, 2, 2, 2]$ in the decoding path. We report results against the same baselines of Section 4.1. Table 5 presents the results in the described setting. We can see that MuFIR manages to outperform all binary methods, effectively denoising degraded images with no multiplication operations. Qualitative results are showcased in Fig. 5.

4.3. Design ablations

In this subsection, we validate several structural decisions made in the design of MuFIR. The results of these ablation experiments on the GoPro dataset are reported in Table 6.

We first assess the impact of weight ternarization on the original NAFNet architecture, without any further modification ("*Naive Tern-NAFNet*"). We remark that this would not eliminate a large number of multiplications due to normalizations, gating, attention and scalings still being present. Still, we notice a drop of about 1.8 dB in PSNR. We attribute some of this drop to sub-optimal scaling of feature values produced by ternary convolutions, which leads to training instability. For this reason, we introduce the extra normalization layers presented in the system architecture. The result of this change coupled with using average absolute deviation instead of standard deviation is reported as "*AAD Tern-NAFNet*". Notice that this experiment does not use APO2 quantization in the normalization layers to ensure a fair evaluation of the importance of extra normalization layers. We see that this design we proposed recovers part of the performance lost by ternarization. Introducing the annealed training procedure produces the "*AAD ann-Tern-NAFNet*" model. This can be directly compared with MuFIR to assess the impact of APO2 quantization alone. We notice that APO2 quantization reduces performance but is needed to achieve a fully multiplication-free design. We also remark that "*AAD ann-Tern-NAFNet*" has slightly lower performance than "*AAD Tern-NAFNet*" due to the annealing schedule being optimized for MuFIR. Fig. 6 displays the training loss progress for "*Naive Tern-NAFNet*", "*AAD Tern-NAFNet*" and "*AAD ann-Tern-NAFNet*",

Table 4
Number of operations performed by MuFIR ($k = 3$) and other tested architectures on the GoPro dataset.

Method	FP32-NAFNet	INT8-NAFNet	BBCU	ReActNet	Bi-Real	BN-Free	IRNet	MuFIR
# FP mults	63.39G	589.81M	302.38M	528.48M	301.99M	679.48M	226.49M	–
# FP adds	63.45G	399.95M	599.79M	1.28G	448.79M	977.27M	222.30M	–
# INT mults	–	63.05G	–	–	–	–	–	**226.49M***
# INT adds	–	63.05G	43.41G	48.17G	43.41G	48.17G	43.41G	**190.31G**
# XNOR ops	–	–	43.49G	48.31G	43.49G	48.32G	43.49G	–
# bitshifts	–	–	–	–	–	–	–	**189.85G**

* Integer multiplications from first and last convolutional layers.

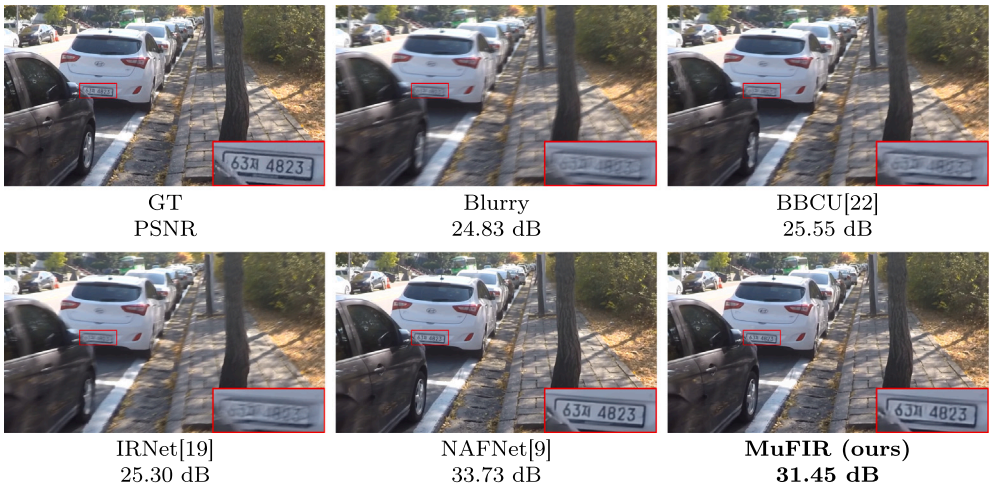


Fig. 4. Qualitative analysis on GoPro. Figures display zoomed-in deblurring details. MuFIR (ours) yields the best visual quality among multiplication-free methods.

Table 5
Quantitative results on the SIDD benchmark for real-world denoising. Metrics are reported in terms of PSNR (dB) and SSIM. Best-performing scores are highlighted in bold.

Method	FP32-NAFNet	INT8-NAFNet	BBCU	ReActNet	Bi-Real	BN-Free	IRNet	MuFIR
PSNR (dB) ↑	40.30	40.09	37.00	29.24	29.74	36.23	36.19	**38.09**
SSIM ↑	0.962	0.961	0.890	0.620	0.659	0.876	0.861	**0.948**

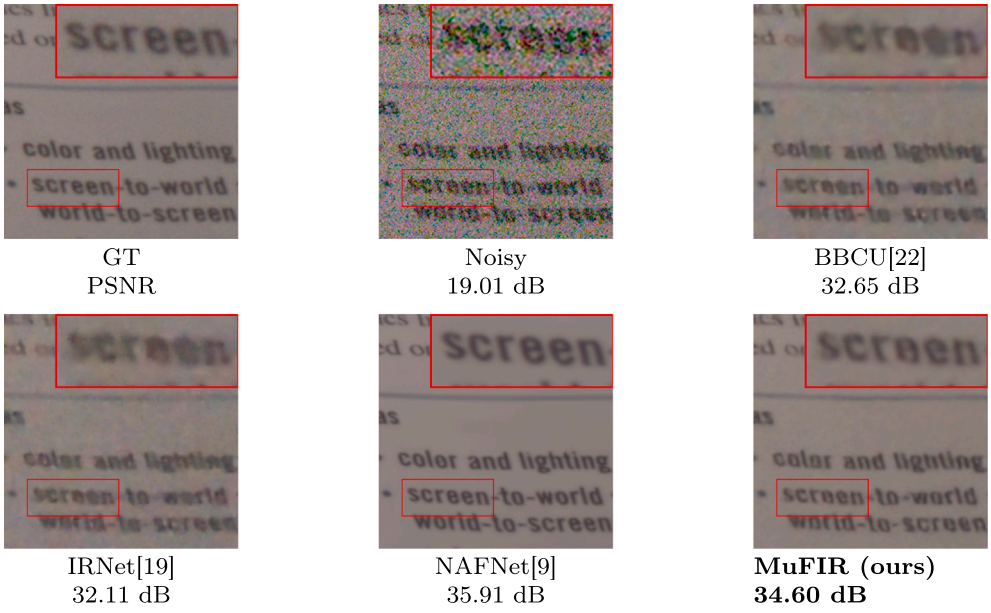


Fig. 5. Qualitative results on SIDD dataset. Top row left to right: ground truth, blurred image, BBCU; bottom row: IRNet, MuFIR, NAFNet.

Table 6
MuFIR experimental ablations on the GoPro dataset. We analyze the effects of ternarization, average absolute deviation and our APO2 quantization. We show that annealed training is necessary with traditional QAT failing.

Method	T	AAD	APO2	Annealed	PSNR(dB)	SSIM
FP32-NAFNet	✗	✗	✗	✗	33.69	0.967
Naive Tern-NAFNet	✓	✗	✗	✗	31.92	0.952
AAD Tern-NAFNet	✓	✓	✗	✗	32.30	0.956
AAD ann-Tern-NAFNet	✓	✓	✗	✓	32.12	0.954
MuFIR QAT	✓	✓	✓	✗	–	–
MuFIR	✓	✓	✓	✓	31.33	0.947

showing that it is more stable with our custom normalization and annealing procedure. We also tested conventional QAT to train MuFIR ("*MuFIR QAT*"), but we do not report the corresponding values for this experiment since it failed to converge to a meaningful performance, highlighting the need for the proposed annealing strategy. This is reflected by the rapidly diverging training-loss curve represented in Fig. 7.

In addition, we report new experiments for hyperparameters $k = 1$ and $k = 5$ in Table 7. Consistent with the scalar error curves, increasing k reduces the approximation error. In particular, $k = 1$ leads to significant drops in PSNR/SSIM, whereas increasing k from 3 to 5 provides only marginal improvements despite requiring more bit-shift operations. This limited gain likely indicates that the number of quantizer levels is already adequate at producing a low approximation error, so further increasing k beyond $k = 3$ yields diminishing returns. Therefore, we consider $k = 3$ to be a good trade-off between reconstruction performance and computational cost, since larger k values increase complexity while offering little additional benefit.

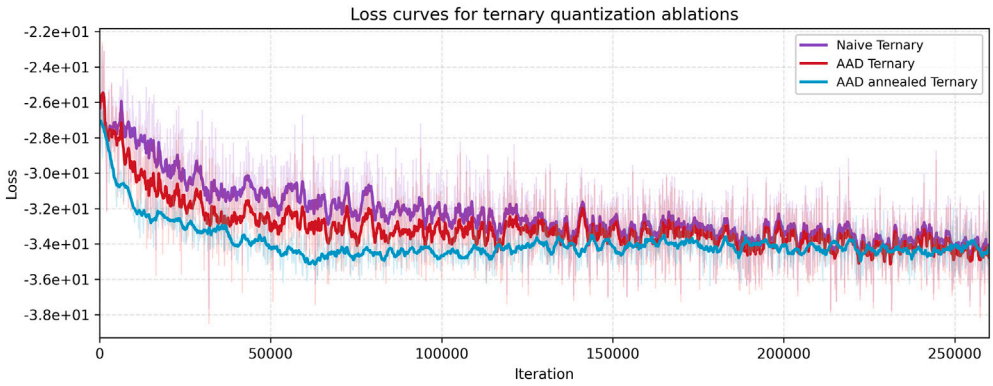


Fig. 6. Ternary ablation loss curves show that AAD norm and annealing stabilize training.

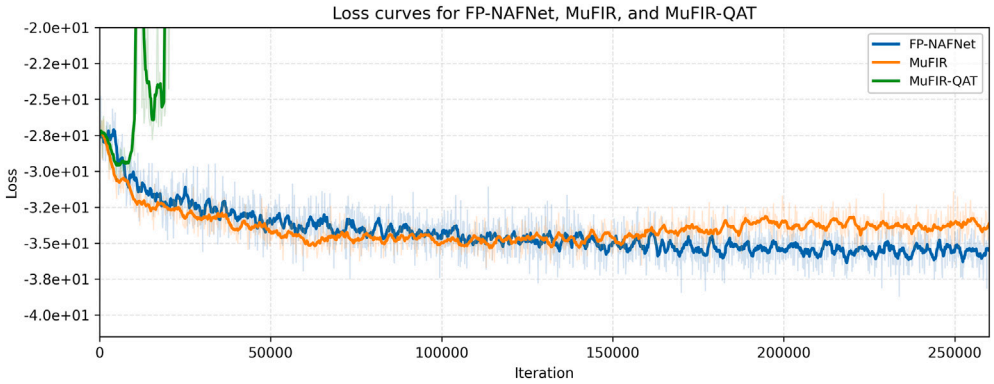


Fig. 7. Loss curves prove the need for the annealing procedure to enable MuFIR training.

Table 7
Ablation on GoPro dataset for hyperparameter k.

Method	MuFIR ($k = 1$)	MuFIR ($k = 3$)	MuFIR ($k = 5$)
PSNR (dB) ↑	26.45	**31.33**	31.42
SSIM ↑	0.871	**0.947**	0.947

5. Conclusions and limitations

We presented a novel neural network backbone for image restoration that entirely avoids multiplication operations thanks to weight ternarization, novel normalization and quantization techniques, and an annealed training strategy for our heavily quantized regime. Even though MuFIR loses in performance compared to full-precision or INT8 architectures, it enforces the much stricter constraint of fully multiplication-free inference, requiring ternary weights, APO2-based bit-shift approximations, and custom normalization throughout the architecture. These constraints naturally introduce a trade-off in restoration performance (e.g., around 2 dB with respect to a simple INT8 architecture), although experimental results on deblurring and denoising show significantly improved performance with respect to non-multiplication-free binary neural networks and close to state-of-the-art models employing multiplications.

The main limitation of our work is that, in order to fully leverage the potential of our multiplication-free design in terms of speed and efficiency improvements, further work is needed on low-level implementations of the proposed techniques, whether with dedicated GPU kernels or ad-hoc hardware.

CRediT authorship contribution statement

Luca Dordoni: Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Diego Valsesia:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Enrico Magli:** Writing – review & editing, Supervision, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data used in this work is publicly available online.

References

[1] C. Dong, C.C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, IEEE Trans. Pattern Anal. Mach. Intell. 38 (2) (2016) 295–307, https://doi.org/10.1109/TPAMI.2015.2439281
[2] J. Kim, J.K. Lee, K.M. Lee, Accurate image super-resolution using very deep convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1646–1654.
[3] K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising, IEEE Trans. Image Process. 26 (7) (2017) 3142–3155.
[4] B. Lim, S. Son, H. Kim, S. Nah, K. Mu Lee, Enhanced deep residual networks for single image super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 136–144.
[5] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, J. Matas, Deblurgan: blind motion deblurring using conditional adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8183–8192.
[6] S.W. Zamir, A. Arora, S. Khan, M. Hayat, F.S. Khan, M.-H. Yang, L. Shao, Multi-stage progressive image restoration, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 14821–14831.
[7] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, R. Timofte, Swinir: image restoration using swin transformer, in: 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), 2021, pp. 1833–1844, https://doi.org/10.1109/ICCVW54120.2021.00210
[8] S.W. Zamir, A. Arora, S. Khan, M. Hayat, F.S. Khan, M.-H. Yang, Restormer: efficient transformer for high-resolution image restoration, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 5728–5739.
[9] L. Chen, X. Chu, X. Zhang, J. Sun, Simple baselines for image restoration, in: European Conference on Computer Vision, Springer, 2022, pp. 17–33.
[10] Y. Bengio, N. Léonard, A. Courville, Estimating or propagating gradients through stochastic neurons for conditional computation, arXiv preprint arXiv:1308.3432, 2013.
[11] M. Courbariaux, Y. Bengio, J.-P. David, Binaryconnect: training deep neural networks with binary weights during propagations, Adv. Neural Inf. Process. Syst. 28 (2015).
[12] Z. Liu, Z. Shen, M. Savvides, K.-T. Cheng, Reactnet: towards precise binary neural network with generalized activation functions, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16, Springer, 2020, pp. 143–159.
[13] Z. Liu, B. Wu, W. Luo, X. Yang, W. Liu, K.-T. Cheng, Bi-real net: enhancing the performance of 1-bit CNNS with improved representational capability and advanced training algorithm, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 722–737.
[14] H. Wang, S. Ma, L. Dong, S. Huang, H. Wang, L. Ma, F. Yang, R. Wang, Y. Wu, F. Wei, Bitnet: Scaling 1-bit transformers for large language models, arXiv preprint arXiv:2310.11453, 2023.
[15] F. Li, B. Liu, X. Wang, B. Zhang, J. Yan, Ternary weight networks, arXiv preprint arXiv:1605.04711, 2016.
[16] C. Zhu, S. Han, H. Mao, W.J. Dally, Trained ternary quantization, arXiv preprint arXiv:1612.01064, 2016.
[17] S. Ma, H. Wang, L. Ma, L. Wang, W. Wang, S. Huang, L. Dong, R. Wang, J. Xue, F. Wei, The era of 1-bit llms: All large language models are in 1.58 bits, arXiv preprint arXiv:2402.17764, 1 2024.
[18] M. Rastegari, V. Ordonez, J. Redmon, A. Farhadi, Xnor-net: imagenet classification using binary convolutional neural networks, in: European Conference on Computer Vision, Springer, 2016, pp. 525–542.
[19] H. Qin, R. Gong, X. Liu, M. Shen, Z. Wei, F. Yu, J. Song, Forward and backward information retention for accurate binary neural networks, in: Proceedings of the

IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 2250–2259.
[20] Y. Ma, H. Xiong, Z. Hu, L. Ma, Efficient super resolution using binarized neural network, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2019.
[21] J. Xin, N. Wang, X. Jiang, J. Li, H. Huang, X. Gao, Binarized neural network for single image super resolution, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16, Springer, 2020, pp. 91–107.
[22] B. Xia, Y. Zhang, Y. Wang, Y. Tian, W. Yang, R. Timofte, L. Van Gool, Basic binary convolution unit for binarized image restoration network, in: ICLR, 2023.
[23] P. Chen, B. Zhuang, C. Shen, Fatnn: fast and accurate ternary neural networks, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 5219–5228.
[24] D.A. Gudovskiy, L. Rigazio, Shiftcnn: Generalized low-precision architecture for inference of convolutional neural networks, arXiv preprint arXiv:1706.02393, 2017.
[25] H. Chen, Y. Wang, C. Xu, B. Shi, C. Xu, Q. Tian, C. Xu, Addernet: do we really need multiplications in deep learning? in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1468–1477.
[26] R.-J. Zhu, Y. Zhang, E. Sifferman, T. Sheaves, Y. Wang, D. Richmond, P. Zhou, J.K. Eshraghian, Scalable matmul-free language modeling, arXiv preprint arXiv:2406.02528, 2024.
[27] D. Song, Y. Wang, H. Chen, C. Xu, C. Xu, D. Tao, Addersr: towards energy efficient image super-resolution, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 15648–15657.
[28] M. Scherer, G. Rutishauser, L. Cavigelli, L. Benini, Cutie: beyond petaop/s/w ternary DNN inference acceleration with better-than-binary energy efficiency, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 41 (4) (2021) 1020–1033.
[29] G. Rutishauser, J. Mihali, M. Scherer, L. Bonini, Xtern: energy-efficient ternary neural network inference on risc-v-based edge systems, in: 2024 IEEE 35th International Conference on Application-Specific Systems, Architectures and Processors (ASAP), IEEE, 2024, pp. 206–213.
[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).
[31] A. Gu, T. Dao, Mamba: Linear-time sequence modeling with selective state spaces, arXiv preprint arXiv:2312.00752, 2023.
[32] S.-J. Cho, S.-W. Ji, J.-P. Hong, S.-W. Jung, S.-J. Ko, Rethinking coarse-to-fine approach in single image deblurring, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 4641–4650.
[33] Y. Shi, B. Xia, X. Jin, X. Wang, T. Zhao, X. Xia, X. Xiao, W. Yang, Vmambair: visual state space model for image restoration, IEEE Trans. Circuits Syst. Video Technol. 35 (6) (2025) 5560–5574.
[34] C. Saharia, J. Ho, W. Chan, T. Salimans, D.J. Fleet, M. Norouzi, Image super-resolution via iterative refinement, IEEE Trans. Pattern Anal. Mach. Intell. 45 (4) (2022) 4713–4726.
[35] B. Xia, Y. Zhang, S. Wang, Y. Wang, X. Wu, Y. Tian, W. Yang, L. Van Gool, Diffir: efficient diffusion model for image restoration, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 13095–13105.
[36] J. Whang, M. Delbracio, H. Talebi, C. Saharia, A.G. Dimakis, P. Milanfar, Deblurring via stochastic refinement, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16293–16303.
[37] X. Mao, Q. Li, Y. Wang, Adarevd: adaptive patch exiting reversible decoder pushes the limit of image deblurring, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 25681–25690.
[38] K. Zhang, W. Zuo, S. Gu, L. Zhang, Learning deep CNN denoiser prior for image restoration, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3929–3938.
[39] R. Li, Y. Wang, S. Chen, F. Zhang, J. Gu, T. Xue, Dualdn: dual-domain denoising via differentiable ISP, in: European Conference on Computer Vision, Springer, 2025, pp. 160–177.
[40] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[41] X. Jiang, N. Wang, J. Xin, K. Li, X. Yang, X. Gao, Training binary neural network without batch normalization for image super-resolution, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, 2021, pp. 1700–1707.
[42] A. Kosson, M. Jaggi, Multiplication-free transformer training via piecewise affine operations, Adv. Neural Inf. Process. Syst. 36 (2023) 8208–8223.
[43] H. You, Y. Guo, Y. Fu, W. Zhou, H. Shi, X. Zhang, S. Kundu, A. Yazdanbakhsh, Y.C. Lin, Shiftaddllm: accelerating pretrained llms via post-training multiplication-less reparameterization, Adv. Neural Inf. Process. Syst. 37 (2024) 24822–24848.
[44] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, D. Kalenichenko, Quantization and training of neural networks for efficient integer-arithmetic-only inference, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2704–2713.
[45] J.L. Ba, J.R. Kiros, G.E. Hinton, Layer normalization, arXiv preprint arXiv:1607.06450, 2016.
[46] X. Li, B. Liu, R.H. Yang, V. Courville, C. Xing, V.P. Nia, Denseshift: towards accurate and efficient low-bit power-of-two quantization, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 17010–17020.
[47] K. Purohit, M. Suin, A.N. Rajagopalan, V.N. Boddeti, Spatially-adaptive image restoration using distortion-guided networks, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 2309–2319.
[48] M. Suin, K. Purohit, A.N. Rajagopalan, Spatially-attentive patch-hierarchical network for adaptive motion deblurring, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 3606–3615.
[49] D. Park, D.U. Kang, J. Kim, S.Y. Chun, Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training, in: European Conference on Computer Vision, Springer, 2020, pp. 327–343.
[50] H. Gao, X. Tao, X. Shen, J. Jia, Dynamic scene deblurring with parameter selective sharing and nested skip connections, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3848–3856.
[51] S. Nah, T. Hyun Kim, K. Mu Lee, Deep multi-scale convolutional neural network for dynamic scene deblurring, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3883–3891.
[52] A. Abdelhamed, S. Lin, M.S. Brown, A high-quality denoising dataset for smartphone cameras, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1692–1700.
[53] T. Chen, Z. Zhang, X. Ouyang, Z. Liu, Z. Shen, Z. Wang, "bnn-bn = ?": training binary neural networks without batch normalization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 4619–4629.

Author biography



Luca Dordoni received the M.Sc. degree in Physics of Complex Systems and the B.Sc. degree in Physical Engineering from the Politecnico di Torino in 2023 and 2020, respectively. He is currently pursuing a Ph.D. degree at the Department of Electronics and Telecommunications (DET), within the Image Processing and Learning group focusing on deep learning methods for image processing, with applications in image compression, efficiency, multimedia and remote sensing domains. His main research interests include neural network quantization and efficient deep learning design, particularly for inverse imaging problems and remote sensing applications.



Diego Valsesia is an Associate Professor with the Department of Electronics and Telecommunications (DET), Politecnico di Torino. His main research interests include remote sensing imaging, and deep learning for inverse problems. He is a Senior Area Editor for the IEEE Transactions on Image Processing, for which he received the 2023 Outstanding Editorial Board Member Award. He is a member of the EURASIP Technical Area Committee for Signal and Data Analytics for Machine Learning and a member of the ELLIS society. He received the IEEE ICIP 2019 Best Paper Award and the IEEE Multimedia 2019 Best Paper Award.



Enrico Magli is a Full Professor at Politecnico di Torino, where he leads the Image Processing and Learning group. He is a Senior Associate Editor of IEEE Journal on Selected Topics in Signal Processing. He is a Fellow of the IEEE and a Fellow of the ELLIS Society for the advancement of AI in Europe. He received the GRSS-IEEE 2011 Transactions Prize Paper Award, the IEEE ICIP 2015 Best Student Paper Award, the IEEE ICIP 2019 Best Paper Award, the IEEE Multimedia 2019 Best Paper Award, and the 2010 and 2014 Best Associate Editor Awards of the IEEE TCSVT.