MR-VNet: Media Restoration using Volterra Networks

Siddharth Roheda, Amit Unde, Loay Rashid
Samsung Research Institute
Bengaluru, India
{sid.roheda, amit.unde, loay.rashid}@samsung.com

Abstract

This research paper presents a novel class of restoration network architecture based on the Volterra series formulation. By incorporating non-linearity into the system response function through higher order convolutions instead of traditional activation functions, we introduce a general framework for image/video restoration. Through extensive experimentation, we demonstrate that our proposed architecture achieves state-of-the-art (SOTA) performance in the field of Image/Video Restoration. Moreover, we establish that the recently introduced Non-Linear Activation Free Network (NAF-NET) can be considered a special case within the broader class of Volterra Neural Networks. These findings highlight the potential of Volterra Neural Networks as a versatile and powerful tool for addressing complex restoration tasks in computer vision.

1. Introduction

In our increasingly visual-oriented world, media such as images and videos play a crucial role in conveying information, expressing emotions, and preserving memories. However, images/videos are susceptible to distortions during the process of capturing (eg. sensor noise, blur, zoom, bad exposure), saving/sharing (eg. compression, down-sampling), and editing (eg. un-natural artefacts). It is crucial to restore such degraded images so as to prevent loss of information and ensure that the best visual quality is delivered to users. This is done through techniques such as image de-noise, de-blur, compression reduction, super-resolution, etc. Recently, Convolutional Neural Networks (CNNs) coupled with ample training data and computational resources has driven remarkable progress in image restoration algorithms [5, 7, 24, 29, 30]. The basic building block in a CNN is a convolutional layer followed by an activation function. The convolution operation provides local connectivity and translational in-variance, while the activation function introduces non-linearity into the network. Following this, Transformer networks that use a more dynamic alternative of self-attention were introduced to resolve some of the shortcomings of CNNs. Namely, instead of relying on convolutional filters that have static weights and cannot adapt to input content, self-attention allows calculation of response at a given pixel by weighted sum of all other positions. The drawback of such methods is that they are extremely heavy and difficult to train. This also makes the analysis and tractability of such networks elusive. More recently, there was an interest shown in activation free networks [4] that are lighter and more powerful than the traditional CNNs. In our work we explore introduction of non-linearity in the network through interaction between the pixels of an image. This is done by performing higher order convolutions to augment linear convolutions. We utilize the well-established Volterra Series [21] to accomplish this task.

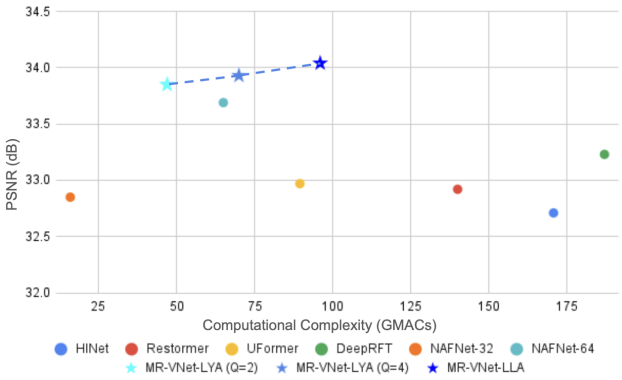

Figure 1. Comparison of PSNR (dB) and Computational Complexity (GMACs) of the various models on the GoPro Dataset.

Contributions: In this paper we propose a novel architecture tailored for image and video restoration that utilizes the recently proposed Volterra layers [18] to optimally introduce non-linearities in the restoration process. We design a U-Net like architecture integrated with Volterra layers to achieve high quality reconstruction while minimizing computational overhead. Our study demonstrates that using the Volterra formulation with the proposed lossless/lossy approximation results in significantly reduced network com-

plexity and depth compared to traditional CNNs to achieve similar performance levels. We also showcase that the recently proposed Non-linear Activation Free NETworks (NAFNet) [4] are a special case of the Volterra Formulation.

2. Related Work

2.1. Volterra Filter

The Volterra Filter, as proposed in [21], serves as an approximation for capturing the non-linear relationship between the input at time t, denoted as x_t, and the corresponding output y_t. Mathematically, this relationship is expressed as follows:

$$y_t = \sum_{\tau_1=0}^{L-1} \boldsymbol{W^1}_{\tau_1} x_{t-\tau_1} + \sum_{\tau_1,\tau_2=0}^{L-1} \boldsymbol{W^2}_{\tau_1,\tau_2} x_{t-\tau_1} x_{t-\tau_2} + ... + \sum_{\tau_1,\tau_2,...,\tau_K=0}^{L-1} \boldsymbol{W^K}_{\tau_1,\tau_2,...,\tau_K} x_{t-\tau_1} x_{t-\tau_2} ... x_{t-\tau_K}, \quad (1)$$

where L represents the filter memory or filter length, and $\boldsymbol{W^K}$ is the weight matrix corresponding to the K^{th} order term. Notably, the computational complexity of this formulation experiences exponential growth with an increase in the desired filter order. Specifically, a K^{th} order filter with length L necessitates solving L^K equations. It is noteworthy that the first term in the equation corresponds to the linear convolutional layer commonly employed in Convolutional Neural Networks (CNNs).

The distinctive feature of Equation 1 is the incorporation of non-linearities through higher order convolutions, as opposed to the traditional activation function. For instance, the second term in Equation 1 specifically refers to a second-order convolution.

2.2. Volterra Filters in Deep Learning

The potential of Volterra Filters in learning non-linear functions is vast, suggesting their potential utility in enhancing the performance of deep learning models. In the study by [31], the authors introduced a second-order Volterra filter to augment non-linearity, supplementing traditional activation functions. The application of a second-order Volterra Filter was also exemplified in facial recognition tasks [9]. Despite demonstrating the efficacy of Volterra Filters, the exploration was constrained to second-order non-linearities, primarily due to computational complexity limitations.

In another study [18], a cascaded approach to Volterra Filtering was proposed to address the challenge of Video Action Recognition. This approach involved cascading layers of second-order filters, resulting in a filter with significantly higher complexity. The research showcased the proficiency of Volterra Filters in learning non-linear information from data, all the while demanding lower computational resources compared to conventional Convolutional Neural Networks that rely on activation functions for introducing non-linearities.

2.3. Image/Video Restoration and Enhancement

Image and video restoration is a crucial task involving the enhancement of images or videos that have suffered degradation during capture or storage, ultimately delivering clean, sharp visuals to enhance user experience. Recent contributions in the literature, as exemplified by [5,11,13,20,29], have leveraged the U-Net architecture [19] and extended its capabilities for performance improvement. However, such extensions often come at the cost of increased model complexity. This escalation in complexity, ranging from the incorporation of multiple U-Net stages to the integration of transformers for image restoration, has resulted in a notable surge in model intricacies for achieving marginal PSNR/SSIM improvements.

In contrast, a Non-Linear Activation Free Network (NAFNet) was introduced in [4], challenging the prevailing notion that high complexities are indispensable. This work argued that a simple baseline network can achieve comparable results. Moreover, empirical evidence from this study demonstrated that non-linear activation functions may be dispensable, substitutable by straightforward element-wise multiplication. This finding aligns with the assertion in [18], where Volterra Filters replaced activation functions for Action Recognition tasks.

On a different front, Video Restoration has garnered increased attention in the research community. Video Restoration tasks present greater complexity than image restoration, given their multi-frame structure and temporal interdependencies among frames. Addressing both spatial and temporal aspects is crucial to ensuring flicker-free restoration and maintaining continuity across video frames. Recent research in Video Restoration, as observed in the use of LSTMs and RNNs [17, 22], emphasizes effective exploitation of temporal information. Alternatively, transformer architectures [10] have been employed to consider frame order and ensure coherence in restored videos. Volterra Filters [21] offer a promising formulation for exploring the temporal dynamics of videos, enabling the generation of non-linear interactions to enhance both spatial and temporal non-linearity.

3. Problem Formulation

Consider a collection of degraded images denoted as $\boldsymbol{X_D} = x_D^1, x_D^2, ..., x_D^N$ of particular interest. The objective is to devise a restoration function, $\boldsymbol{G} : \mathbb{R}^{H\times W\times C} \rightarrow \mathbb{R}^{H\times W\times C}$, capable of recovering a set of clean images, $\boldsymbol{X_R} = x_R^1, x_R^2, ..., x_R^N$ from $\boldsymbol{X_D}$. This restorative func-

tion is structured as a composition of an encoder, a middle block, and a decoder, expressed as follows:

$$\boldsymbol{G} = \boldsymbol{\mathcal{F}_D} \circ \boldsymbol{\mathcal{F}_M} \circ \boldsymbol{\mathcal{F}_E}. \tag{2}$$

4. Proposed Solution

4.1. Volterra Filters for Restoration

In our proposed methodology, we employ the second-order Volterra formulation to implement a Volterra layer. Specifically, the z^{th} layer of the Volterra Neural Network (VNN) processes the input X_{z-1} as follows:

$$X_z = V_z(X_{z-1}) = \boldsymbol{\mathcal{F}_z^1}(X_{z-1}) + \boldsymbol{\mathcal{F}_z^2}(X_{z-1}), \tag{3}$$

where $\boldsymbol{\mathcal{F}_z^1}$ and $\boldsymbol{\mathcal{F}_z^2}$ represent the first and second-order convolution operations in the z^{th} layer. For video data tasks, a spatio-temporal (3D) version of the Volterra Filter is employed to compute the feature value at location $[t, m_1, m_2]$, given by Equation 4:

$$\begin{aligned}\boldsymbol{X}_{z\begin{bmatrix} t \\ m_1 \\ m_2 \end{bmatrix}} &= V_z\left(\boldsymbol{X}_{z-1\begin{bmatrix} t-L:t+L \\ m_1-p_1:m_1+p_1 \\ m_2-p_2:m_2+p_2 \end{bmatrix}}\right) \\ &= \sum_{\tau_1,\sigma_{11},\sigma_{21}} \boldsymbol{W^1}_{\begin{bmatrix} \tau_1 \\ \sigma_{11} \\ \sigma_{21} \end{bmatrix}} x_{\begin{bmatrix} t-\tau_1 \\ m_1-\sigma_{11} \\ m_2-\sigma_{21} \end{bmatrix}} \\ &+ \sum_{\substack{\tau_1,\sigma_{11},\sigma_{21} \\ \tau_2,\sigma_{12}\sigma_{22}}} \boldsymbol{W^2}_{\begin{bmatrix} \tau_1 \\ \sigma_{11} \\ \sigma_{21} \end{bmatrix}\begin{bmatrix} \tau_2 \\ \sigma_{12} \\ \sigma_{22} \end{bmatrix}} x_{\begin{bmatrix} \tau_1 \\ m_1-\sigma_{11} \\ m_2-\sigma_{21} \end{bmatrix}} x_{\begin{bmatrix} t-\tau_2 \\ m_1-\sigma_{12} \\ m_2-\sigma_{22} \end{bmatrix}},\end{aligned} \tag{4}$$

where $\tau_i \in [-L, L]$, $\sigma_{1j} \in [-p_1, p_1]$, and $\sigma_{2j} \in [-p_2, p_2]$ represent temporal and spatial translations (horizontal and vertical directions), respectively. Notably, the first term corresponds to the linear convolution utilized in Convolutional Neural Networks (CNNs), while the second term introduces non-linearity in the network instead of relying on a conventional activation function.

For image data, where only spatial translations (2D) are applicable, the feature value at location $[m_1, m_2]$ in feature map X_z is computed using a 2D version of the Volterra Filter, as expressed in Equation 5:

$$\begin{aligned}\boldsymbol{X}_{z\begin{bmatrix} m_1 \\ m_2 \end{bmatrix}} &= V_z\left(\boldsymbol{X}_{z-1\begin{bmatrix} m_1-p_1:m_1+p_1 \\ m_2-p_2:m_2+p_2 \end{bmatrix}}\right) \\ &= \sum_{\sigma_{11},\sigma_{21}} \boldsymbol{W^1}_{\begin{bmatrix} \sigma_{11} \\ \sigma_{21} \end{bmatrix}} x_{\begin{bmatrix} m_1-\sigma_{11} \\ m_2-\sigma_{21} \end{bmatrix}} \\ &+ \sum_{\substack{\sigma_{11},\sigma_{21} \\ \sigma_{12}\sigma_{22}}} \boldsymbol{W^2}_{\begin{bmatrix} \sigma_{11} \\ \sigma_{21} \end{bmatrix}\begin{bmatrix} \sigma_{12} \\ \sigma_{22} \end{bmatrix}} x_{\begin{bmatrix} m_1-\sigma_{11} \\ m_2-\sigma_{21} \end{bmatrix}} x_{\begin{bmatrix} m_1-\sigma_{12} \\ m_2-\sigma_{22} \end{bmatrix}},\end{aligned} \tag{5}$$

where $\sigma_{1j} \in [-p_1, p_1]$ and $\sigma_{2j} \in [-p_2, p_2]$ represent spatial translations in the horizontal and vertical directions, respectively.

We incorporate these Volterra filter layers in a cascaded fashion, following the simple U-Net architecture. The cascading of the layers as defined in Equations 4 and 5 leads to the approximation of a higher-order Volterra filter.

Proposition 1. [Higher Order Complexity] [18] If $\mathcal{Z}$ second-order filters are cascaded, the resulting Volterra Network achieves an effective order of $\mathcal{K}_z = 2^{2^{\mathcal{Z}-1}}$.

The implementation of a K^{th} order Volterra Filter, as specified in Equation 1, necessitates $\sum_{k=1}^{K}(L.\mathcal{P}_1.\mathcal{P}_2)^k$ parameters. Where $\mathcal{P}_1 = 2p_1 + 1$, $\mathcal{P}_2 = 2p_2 + 1$ and $L = 1$ in case of images. The adoption of cascaded 2^{nd} order filters emerges as a more resource-efficient approach for implementing the Volterra formulation, significantly reducing the required parameters [18].

Proposition 2. [VNN Complexity] [18] The complexity of a cascaded $\mathcal{K}_z^{th}$ order Volterra filter is given by:

$$\sum_{z=1}^{\mathcal{Z}} \left[(L_z.\mathcal{P}_{1_z}\mathcal{P}_{2_z} + (L_z.\mathcal{P}_{1_z}\mathcal{P}_{2_z})^2)\right]. \tag{6}$$

As previously mentioned, our approach eschews explicit activation functions like ReLU, tanh, etc. The non-linearity is learned as a function of the input data. Proposition 3 highlights that the VNN formulation is proficient in approximating well-known activation functions, implying that VNN offers a generalized activation encompassing ReLU, sigmoid, tanh, etc., as special cases.

Proposition 3. [Generalized Activation] A Volterra Filter, as described in Equation 1, provides an approximation to any continuous function.

Proof. Employing the Taylor expansion, any non-linear function, $\sigma(.)$, can be expressed as:

$$\sigma(x) = c^0 + c^1 x + ... + c^K x^K + ...c^\infty x^\infty. \tag{7}$$

Specifically, a Sigmoid can be written as:

$$\sigma_{sigmoid}(x) = \frac{1}{1+e^x} = \frac{1}{2} + \frac{1}{4}x - \frac{1}{48}x^2 + \frac{1}{480}x^5... \tag{8}$$

From Equation 1, the Volterra Filter formulation can effectively approximate such an expansion upto a finite order when $L = p_1 = p_2 = 1$,

$$\sigma_{VF} = w^0 + w^1 x + w^2 x^2 + ... + w^K x^K, \tag{9}$$

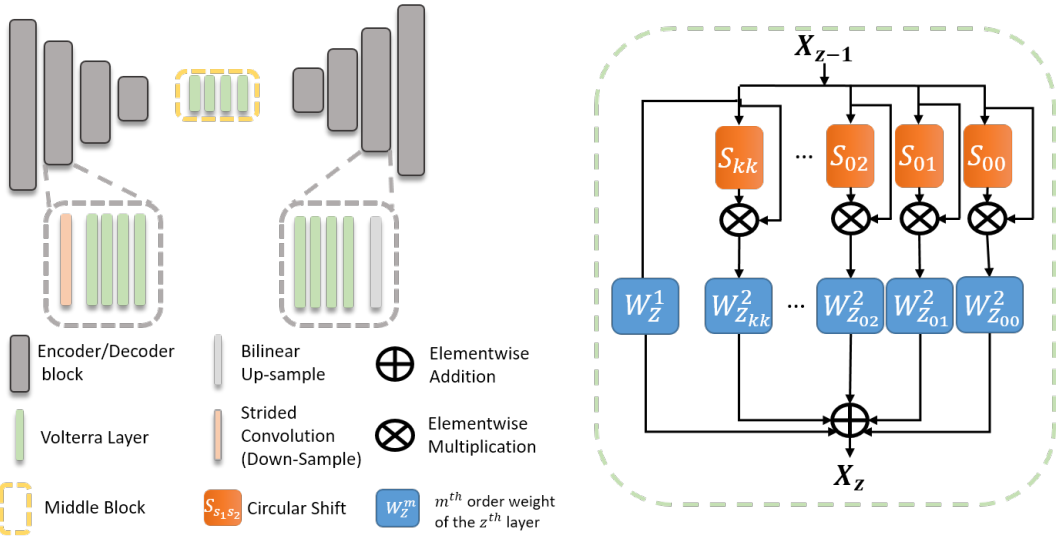

Figure 2. High level Block Diagram for implementation of the proposed VNN Image Restoration Model

where w^k is learned during the training process, rendering it a more generalized activation function capable of learning data-dependent non-linearities. When w^k in Equation 9 is assigned the values corresponding to the Taylor expansion coefficients of a specific activation function (such as the sigmoid in Equation 8), it assumes the role of a predefined activation function, representing a special case within the broader framework.

4.2. Realization of Higher Order Convolutions

4.2.1 Lossless Approximation

We will now elaborate on the implementation of the higher-order kernels outlined in Equation 3. The Volterra Formulation from Equation 3 can be expressed as follows:

$$X_z = \boldsymbol{W_z^1} \star X_{z-1} + \sum_{s_1,s_2} \boldsymbol{W_{z_{s_1,s_2}}^2} \star X_{z-1}.\mathcal{S}_{s_1,s_2}(X_{z-1}), \tag{10}$$

Here, s_1 and s_2 denote spatial shifts in the input feature map X_{z-1}, and $\mathcal{S}_{s_1,s_2}$ represents the feature map circularly shifted along its rows and columns by s_1 and s_2 respectively. The formulation described in Equation 10 enables the parallel implementation of second order Volterra kernel using 2D convolutions as part of PyTorch [16] or Tensorflow [1].

Consider the scenario where $\mathcal{P}_1 = \mathcal{P}_2 = \mathcal{P}$. To implement the second term in Equation 10, $\mathcal{P}^2$ convolutions would be required. To mitigate redundancy and model complexity, we discard the symmetric terms, resulting in ${}^{\mathcal{P}}C_2 < \mathcal{P}^2$ convolutions. This approach reduces the model's complexity by eliminating redundant information, thus serving as a lossless approximation of the exact kernel. In the subsequent section, we delve into a lossy approximation for the second-order Volterra kernel.

4.2.2 Lossy Approximation

Implementing higher-order convolution can incur significant costs despite the lossless optimization techniques discussed in Section 4.2.1. To address this, we employ the concept of separable kernels to approximate the second-order convolution. In the 2D case, the second-order filter is realized as follows:

$$\boldsymbol{W}^{\mathbf{2}}_{\mathcal{P}_1\times\mathcal{P}_2\times\mathcal{P}_1\times\mathcal{P}_2} = \sum_{q=1}^{Q} \boldsymbol{W}^{\mathbf{2}}_{aq_{\mathcal{P}_1\times\mathcal{P}_2\times 1}} \boldsymbol{W}^{\mathbf{2}}_{bq_{1\times\mathcal{P}_1\times\mathcal{P}_2}}, \tag{11}$$

where Q is the desired rank of approximation, $\mathcal{P}_1 = 2p_1 + 1$ and $\mathcal{P}_2 = 2p_2 + 1$. A similar separable kernel approach was explored in [3]. However, authors of [3] only considered a 1^{st} rank ($Q = 1$) approximation, leading to sub-optimal performance as detailed in Tables 5 and 6.

With this approximation, the Volterra Net Block can be implemented as follows:

$$\boldsymbol{W_z^1} \star X_{z-1} + \sum_{q=1}^{Q} \boldsymbol{W_{aq}^2} \star X_{z-1}.\boldsymbol{W_{bq}^2} \star X_{z-1}. \tag{12}$$

This reduces the complexity of network as $\sum_{z=1}^{Z}\left[(L_z.\mathcal{P}_{1_z}.\mathcal{P}_{2_z}) + 2Q(L_z.\mathcal{P}_{1_z}.\mathcal{P}_{2_z})\right]$, which is significantly lower than Equation 6.

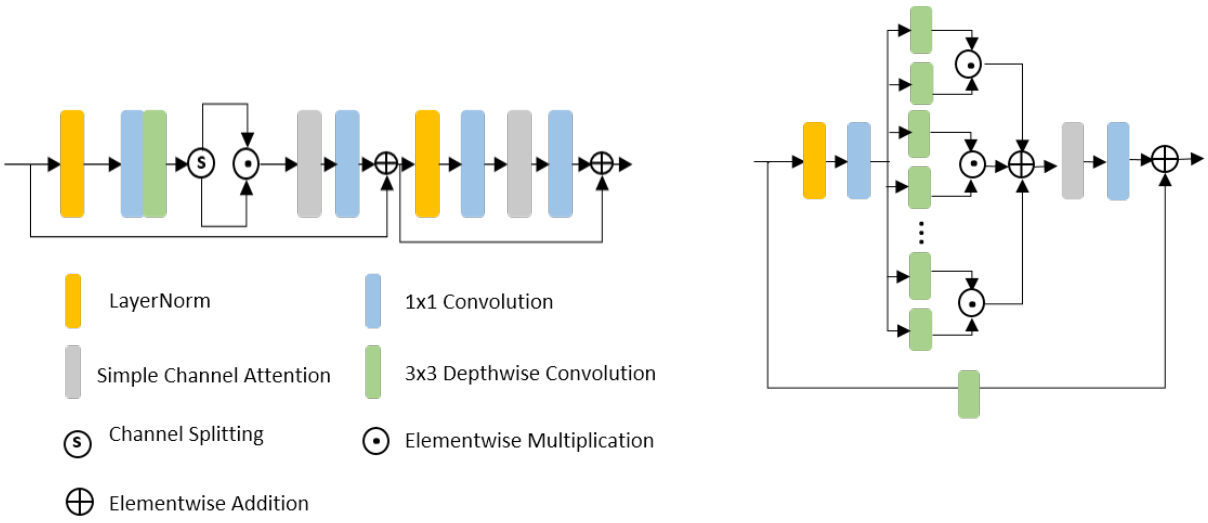

Figure 3. NAFNet Block on the left, MR-VNN block (lossy approximation) on the right

Proposition 4. [Second Order Approximation] The lossy approximation in Equation 11 is a Q^{th} rank approximation of the exact quadratic kernel.

Proof. Let's examine a 1-D Volterra Filter with length L. The quadratic weight matrix, $\boldsymbol{W^2}$, in such a case has dimension $L \times L$, and Equation 11 becomes:

$$\boldsymbol{W^{2\,(Q)}_{L\times L}} = \sum_{q=1}^{Q} \boldsymbol{W^2_{a_{qL\times 1}}} \boldsymbol{W^2_{b_{q1\times L}}}. \tag{13}$$

Singular Value Decomposition of $\boldsymbol{W^2}$ leads to:

$$\boldsymbol{W^2} = \boldsymbol{U\Sigma V}^T, \tag{14}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are $L \times L$ matrices, and $\boldsymbol{\Sigma}$ is a diagonal matrix with singular values of $\boldsymbol{W^2}$. Consequently, Equation 14 becomes:

$$\boldsymbol{W^2} = \sum_{q=1}^{L} u_q \sigma_q v_q^T, \tag{15}$$

where the q^{th} basis of $\boldsymbol{U}$ and $\boldsymbol{V}$ are represented by u_q and v_q respectively, and σ_q is the q^{th} singular value of $\boldsymbol{W^2}$. A Q^{th} rank approximation is then given as:

$$\boldsymbol{W^2(Q)} = \sum_{q=1}^{Q} u_q \sigma_q v_q^T \tag{16}$$

$$= \sum_{q=1}^{Q} \hat{u}_q v_q, \tag{17}$$

where $\hat{u}_q = u_q.\sigma_q$. If $\boldsymbol{W^2_{aq}} = \hat{u}_q$ and $\boldsymbol{W^2_{bq}} = v_q^T$,

$$\boldsymbol{W^{2(Q)}} = \sum_{q=1}^{Q} \boldsymbol{W^2_{aq}} \boldsymbol{W^2_{bq}}. \tag{18}$$

This proves that the approximation detailed in Equation 11 is indeed a Q^{th} rank approximation of the exact quadratic kernel, $\boldsymbol{W^2}$.

Proposition 5. [Special Case of NAFNet] The Non-Linear Activation Free Net (NAFNet) is a special case of Volterra Neural Networks, characterized by $\boldsymbol{W^1_z} = \beta.\boldsymbol{I}$ and $Q = 1$.

Proof. Consider the tensor $X_{z-1_{W\times H\times C}}$ to be the input to the z^{th} NAF-Block. The NAF-Block processes the input by convolving it with a kernel, $W_{z_{NAF}}$, yielding the intermediate output, $L_{z_{W\times H\times 2C}} = W_{z_{NAF}} \star X_{z-1}$. Subsequently, L_z is split into $L_{za_{W\times H\times C}}$ and $L_{zb_{W\times H\times C}}$, which are multiplied to produce the simple gate output, $M_z = L_{za}.L_{zb}$. Finally, the NAF-Block output is obtained as $X_z = M_z + \beta X_{z-1}$, where β is a scaling factor. As $W_{z_{NAF}}$ is a depthwise convolution with $groups = 2c$, it can be expressed as the product of separable convolutions:

$$\boldsymbol{W}_{z_{NAF}} = \boldsymbol{W}_{za_{NAF}}.\boldsymbol{W}_{zb_{NAF}}, \tag{19}$$

Consequently, M_z is computed as,

$$\begin{aligned} M_z &= \boldsymbol{W}_{za_{NAF}} \star X_{z-1}.\boldsymbol{W}_{zb_{NAF}} \star X_{z-1} \\ &= L_{za}.L_{zb}. \end{aligned} \tag{20}$$

Setting $Q = 1$ in Equation 11 yields the expression in Equation 20, demonstrating that the simple gate used in NAFNet Blocks is a 1^{st} rank approximation of the quadratic Volterra kernel.

Method	MIMO-UNet [7]	HINet [5]	MAXIM [20]	Restormer [27]	UFormer [23]	Deep-RFT [13]	MPR-Net [8]	NAF-Net [4]	MR-VNet-LYA (Q=2)	MR-VNet-LYA (Q=4)	MR-VNet-LLA
PSNR	32.68	32.71	32.86	32.92	32.97	33.23	33.31	33.69	33.85	<u>33.93</u>	**34.04**
SSIM	0.959	0.959	0.961	0.961	<u>0.967</u>	0.963	0.964	<u>0.967</u>	<u>0.967</u>	<u>0.967</u>	**0.969**
GMACs	1235	170.7	169.5	140	89.5	187	778.2	<u>65</u>	**47**	70	96

Table 1. De-Blurring Performance on GoPro Dataset. Best results are **bold**, second best are <u>underlined</u>.

Method	MPR-Net [8]	MIR-Net [28]	NBNet [6]	UFormer [23]	MAXIM [20]	HINet [5]	Restormer [27]	NAF-Net [4]	MR-VNet-LYA (Q=2)	MR-VNet-LYA (Q=4)	MR-VNet-LLA
PSNR	39.71	39.72	39.75	39.89	39.96	39.99	40.02	40.30	40.39	<u>40.43</u>	**40.58**
SSIM	0.958	0.959	0.959	0.960	0.960	0.958	0.960	<u>0.962</u>	<u>0.962</u>	**0.963**	**0.963**
GMACs	588	786	88.8	89.5	169.5	170.7	140	<u>65</u>	**47**	70	96

Table 2. De-Noising Performance on SIDD Dataset. Best results are **bold**, second best are <u>underlined</u>.

Furthermore, a residual is added to the simple gate output to obtain the final output, which can be written as $X_z = M_z + \beta.\boldsymbol{I} \star X_{z-1}$. Consequently, the NAF-Block emerges as a special case of the Volterra Networks with $\boldsymbol{W_z^1} = \beta.\boldsymbol{I}$ and $Q = 1$.

The comparison of a NAFNet Block and a MR-VNet (Lossy Approximation) Block is depicted in Figure 3

4.3. Model Architecture

We design a U-Net inspired architecture for approximating the Encoder-Decoder functions. The input image undergoes initial processing through the Encoder function, $\boldsymbol{\mathcal{F}_E} : \mathbb{R}^{W\times H\times C} \rightarrow \mathbb{R}^{w\times h\times c}$, comprising 4 encoder blocks. Each encoder block consists of 4 Volterra Layers, implemented as described by Equation 5 utilizing the Lossless and Lossy approximations from Sections 4.2.1 and 4.2.2. Strided convolutions are employed in consecutive Encoder blocks to reduce resolution, resulting in a latent space configuration of $h = H/8$ and $w = W/8$. The middle block in the latent space, $\boldsymbol{\mathcal{F}_M} : \mathbb{R}^{h\times w\times c} \rightarrow \mathbb{R}^{h\times w\times c}$, is implemented using a single Volterra Layer. Finally, the decoder function, $\boldsymbol{\mathcal{F}_D} : \mathbb{R}^{h\times w\times c} \rightarrow \mathbb{R}^{H\times W\times C}$, transforms the encoded features back into the image space. The Decoder, symmetric in design to the Encoder with 4 blocks, has only 1 Volterra Layer per block. Each encoder block is connected to its corresponding decoder block via a residual connection. For video restoration problems, the same architecture is retained, but 3D Volterra Filters are used instead of 2D, as detailed in Equation 4. The entire architectural arrangement is illustrated in Figure 2.

5. Experiments

To assess the efficacy of the proposed Volterra Restoration Network, we conduct experiments targeting prevalent degradations in images and videos:

- **Motion-Blur**: This degradation, arising from camera or subject motion, is addressed by training and testing the restoration network on the GoPro [14] and Reds [15] datasets for image deblurring.
- **Camera Sensor Noise**: We aim to mitigate noise introduced by the camera sensor during image/video capture. Evaluation is performed using the SIDD [2] and CRVD [25] datasets.

We present two iterations of our proposed Volterra Layer-based architecture: MR-VNet-LYA, employing the Lossy approximation discussed in Section 4.2.2, and MR-VNet-LLA, incorporating the proposed Lossless approximation. Subsequently, we subject our method to a comprehensive evaluation against state-of-the-art (SOTA) algorithms in the domains of Image and Video Restoration. Evaluation metrics such as Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) are employed for a thorough assessment. Additionally, we conduct a detailed comparative analysis of the computational complexities associated with the proposed methods, quantified in terms of Giga Multiply-Add Computations (GMACs).

The efficacy of our proposed method is substantiated through comparisons with various SOTA techniques prevalent in the literature. A detailed quantitative assessment for De-Blurring is presented in Tables 1 and 3, utilizing the GoPro and REDS datasets, respectively. The results underscore the superior performance of our proposed model, outperforming existing SOTA methods in terms of both PSNR and SSIM metrics.

In the realm of image denoising, our proposed method outperforms alternative approaches, as evidenced by the quantitative results presented in Table 2 on the SIDD dataset. Notably, our technique demonstrates superior de-

Figure 4. De-Blurring Results on GoPro.

Figure 5. De-Noising Results on SIDD. Left to Right: Noisy Image, NAFNet, MR-VNet (Ours)

Method	PSNR	SSIM	GMACs
MPRNet [8]	28.79	0.811	776.7
HiNet [5]	28.83	0.862	170.7
MAXIM [20]	28.93	0.865	169.5
NAFNet [4]	29.09	0.867	65
MR-VNet-LYA (Q=4)	<u>29.79</u>	<u>0.868</u>	70
MR-VNet-LLA	**29.92**	**0.869**	96

Table 3. De-Blurring Performance on REDS Dataset

noising efficacy, as reflected in higher Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) scores compared to competing methods.

Furthermore, our exploration extends to the challenging domain of video denoising, detailed in Table 4. In this scenario, we utilize 3D convolution as detailed in Equation 4 and exclusively assess the performance of our Lossy approximation due to its practical efficiency compared to the computationally demanding exact kernel. Our results underscore the effectiveness of the proposed method in achieving superior denoising outcomes in video data, showcasing its competitive edge over existing methodologies.

Method	PSNR	SSIM	GMACs
EMVD-L [12]	41.00	0.983	2543
RVIDEFormer [26]	41.29	0.984	287
LLVD-L [17]	<u>41.41</u>	<u>0.984</u>	117
MR-VNet-LYA	**<u>41.93</u>**	**<u>0.985</u>**	163

Table 4. De-Noising Performance on CRVD Dataset. Best performance is **bold**, second best is <u>underlined</u>.

Kernel Rank	PSNR	SSIM	GMACs
Q=1 (Special case of NAFNet)	33.50	0.965	36
Q=2	33.85	0.966	47
Q=4	33.93	0.967	70
Q=8	33.97	0.967	115

Table 5. Effect of the rank of 2^{nd} order Kernel for GoPro

Kernel Rank	PSNR	SSIM	GMACs
Q=1 (Special case of NAFNet)	40.21	0.962	36
Q=2	40.39	0.963	47
Q=4	40.43	0.963	70
Q=8	40.46	0.964	115

Table 6. Effect of the rank of 2^{nd} order Kernel for SIDD

These compelling quantitative results affirm the efficacy of our proposed method in addressing both image and video denoising challenges, positioning it as a robust and advanced solution in comparison to contemporary approaches.

5.1. Ablation Study

In our comprehensive ablation study, we rigorously analyze the individual contributions of various components within the proposed architecture.

We first investigate the impact of selecting the rank of the kernel in the context of lossy approximation, as demonstrated on the GoPro and SIDD datasets, as detailed in Tables 5 and 6. Notably, when the kernel is constrained to rank 1, it aligns with the specific case observed in NAFNet. As we escalate the rank, the approximation becomes progressively accurate, converging towards the precision of the exact kernel. However, this refinement comes at the expense of heightened complexity.

In Table 7, we demonstrate the impact of width of the architecture in terms of channels while keeping the quadratic filter rank constant.

In Table 8, we present the impact of incorporating activation functions alongside the 2^{nd} order kernel. Our observations indicate that, while ReLU and Sigmoid activation confer a marginal improvement in the case of NAFNet (1^{st} Rank approximation), the same effect is not discernible when employing a higher rank approximation of the Volterra Filter. This suggests that the 1^{st} Rank kernel utilized in NAFNet is insufficient to closely approximate the exact quadratic kernel. Consequently, the activation function introduces some non-linearity to the model, resulting in improved performance. However, with a closer approximation in the form of a higher rank kernel, none of the activation functions appear to contribute additional non-linearity beyond what the model has already learned.

Width	PSNR	SSIM	GMACs
16	32.80	0.961	8
24	32.92	0.963	18
32	33.18	0.966	32
48	33.93	0.967	70

Table 7. Effect of width of the architecture when Q=4 (GoPro Dataset)

Activation	NAFNet	MR-VNet
Identity	39.96	40.43
ReLU	39.98	40.40
GELU	39.97	40.39
Sigmoid	39.99	40.40
SiLU	39.96	40.39

Table 8. Impact of using activation functions. The performance is evaluated in terms of PSNR on the SIDD Dataset

6. Conclusion

In conclusion, our research introduces the Media Restoration-Volterra Network (MR-VNet) as a novel approach to image and video restoration. Leveraging higher-order Volterra filters, the proposed architecture demonstrates promising capabilities in addressing common image and video degradations, such as motion blur and camera sensor noise. Through the development of two architectural variants, VNN-LYA and VNN-LLA, employing lossy and lossless approximations, respectively, we offer a comprehensive exploration of the network's performance.

Our experimental evaluations, conducted on diverse datasets including GoPro, Reds, SIDD, and CRVD, showcase the effectiveness of MR-VNet in comparison to state-of-the-art algorithms. Notably, MR-VNet exhibits competitive results in terms of Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM). Furthermore, we analyze the impact of activation functions on the network's performance, revealing insights into their efficacy in conjunction with higher-order Volterra filters.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016. 4

[2] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1692–1700, 2018. 6

[3] Monami Banerjee, Rudrasis Chakraborty, Jose Bouza, and Baba C Vemuri. Volterranet: A higher order convolutional network with group equivariance for homogeneous manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(2):823–833, 2020. 4

[4] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *European Conference on Computer Vision*, pages 17–33. Springer, 2022. 1, 2, 6, 7

[5] Liangyu Chen, Xin Lu, Jie Zhang, Xiaojie Chu, and Chengpeng Chen. Hinet: Half instance normalization network for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 182–192, 2021. 1, 2, 6, 7

[6] Shen Cheng, Yuzhi Wang, Haibin Huang, Donghao Liu, Haoqiang Fan, and Shuaicheng Liu. Nbnet: Noise basis learning for image denoising with subspace projection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4896–4906, 2021. 6

[7] Sung-Jin Cho, Seo-Won Ji, Jun-Pyo Hong, Seung-Won Jung, and Sung-Jea Ko. Rethinking coarse-to-fine approach in single image deblurring. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4641–4650, 2021. 1, 6

[8] Xiaojie Chu, Liangyu Chen, Chengpeng Chen, and Xin Lu. Improving image restoration by revisiting global information aggregation. In *European Conference on Computer Vision*, pages 53–71. Springer, 2022. 6, 7

[9] Ritwik Kumar, Arunava Banerjee, Baba C Vemuri, and Hanspeter Pfister. Trainable convolution filters and their application to face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1423–1436, 2011. 2

[10] Jingyun Liang, Jiezhang Cao, Yuchen Fan, Kai Zhang, Rakesh Ranjan, Yawei Li, Radu Timofte, and Luc Van Gool. Vrt: A video restoration transformer. *arXiv preprint arXiv:2201.12288*, 2022. 2

[11] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1833–1844, 2021. 2

[12] Matteo Maggioni, Yibin Huang, Cheng Li, Shuai Xiao, Zhongqian Fu, and Fenglong Song. Efficient multi-stage video denoising with recurrent spatio-temporal fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3466–3475, 2021. 8

[13] Xintian Mao, Yiming Liu, Fengze Liu, Qingli Li, Wei Shen, and Yan Wang. Intriguing findings of frequency selection for image deblurring. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 1905–1913, 2023. 2, 6

[14] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3883–3891, 2017. 6

[15] Seungjun Nah, Sanghyun Son, Suyoung Lee, Radu Timofte, and Kyoung Mu Lee. Ntire 2021 challenge on image deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 149–165, 2021. 6

[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 4

[17] Loay Rashid, Siddharth Roheda, and Amit Unde. Llvd: Lstm-based explicit motion modeling in latent space for video denoising. 2, 8

[18] Siddharth Roheda and Hamid Krim. Conquering the cnn over-parameterization dilemma: A volterra filtering approach for action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11948–11956, 2020. 1, 2, 3

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 2

[20] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxim: Multi-axis mlp for image processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5769–5780, 2022. 2, 6, 7

[21] Vito Volterra. Theory of functionals and of integral and integro-differential equations. *(No Title)*, 1959. 1, 2

[22] Wei Wang, Xin Chen, Cheng Yang, Xiang Li, Xuemei Hu, and Tao Yue. Enhancing low light videos by exploring high sensitivity camera noise. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4111–4119, 2019. 2

[23] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17683–17693, 2022. 6

[24] Qingyang Xu, Chengjin Zhang, and Li Zhang. Denoising convolutional neural network. In *2015 IEEE International Conference on Information and Automation*, pages 1184–1187. IEEE, 2015. 1

[25] Huanjing Yue, Cong Cao, Lei Liao, Ronghe Chu, and Jingyu 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*, pages 2301–2310, 2020. 6

[26] Huanjing Yue, Cong Cao, Lei Liao, and Jingyu Yang. Rvideformer: Efficient raw video denoising transformer with a larger benchmark dataset. *arXiv preprint arXiv:2305.00767*, 2023. 8

[27] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5728–5739, 2022. 6

[28] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Learning enriched features for real image restoration and enhancement. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pages 492–511. Springer, 2020. 6

[29] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14821–14831, 2021. 1, 2

[30] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3929–3938, 2017. 1

[31] Georgios Zoumpourlis, Alexandros Doumanoglou, Nicholas Vretos, and Petros Daras. Non-linear convolution filters for cnn-based learning. In *Proceedings of the IEEE international conference on computer vision*, pages 4761–4769, 2017. 2