Simple Baselines for Image Restoration

Liangyu Chen⋆(✉), Xiaojie Chu⋆, Xiangyu Zhang, and Jian Sun

MEGVII Technology, Beijing, CN
{chenliangyu,chuxiaojie,zhangxiangyu,sunjian}@megvii.com

Abstract. Although there have been significant advances in the field of image restoration recently, the system complexity of the state-of-the-art (SOTA) methods is increasing as well, which may hinder the convenient analysis and comparison of methods. In this paper, we propose a simple baseline that exceeds the SOTA methods and is computationally efficient. To further simplify the baseline, we reveal that the nonlinear activation functions, e.g. Sigmoid, ReLU, GELU, Softmax, etc. are not necessary: they could be replaced by multiplication or removed. Thus, we derive a Nonlinear Activation Free Network, namely NAFNet, from the baseline. SOTA results are achieved on various challenging benchmarks, e.g. 33.69 dB PSNR on GoPro (for image deblurring), exceeding the previous SOTA 0.38 dB with only 8.4% of its computational costs; 40.30 dB PSNR on SIDD (for image denoising), exceeding the previous SOTA 0.28 dB with less than half of its computational costs. The code and the pre-trained models are released at github.com/megvii-research/NAFNet.

Keywords: Image Restoration, Image Denoise, Image Deblur

1 Introduction

With the development of deep learning, the performance of image restoration methods improve significantly. Deep learning based methods[4,35,37,34,5,6,30,7,24] have achieved tremendous success. E.g. [37] and [7] achieve 40.02/33.31 dB of PSNR on SIDD[1]/GoPro[25] for image denoising/deblurring respectively.

Despite their good performance, these methods suffer from high system complexity. For a clear discussion, we decompose the system complexity into two parts: inter-block complexity and intra-block complexity. First, the inter-block complexity, as shown in Figure 2. [6,24] introduce connections between various-sized feature maps; [4,35] are multi-stage networks and the latter stage refine the results of the previous stage. Second, the intra-block complexity, i.e. the various design choices inside the block. E.g. Multi-Dconv Head Transposed Attention Module and Gated Dconv Feed-Forward Network in [37] (as we shown in Figure 3a), Swin Transformer Block in [21], HINBlock in [4], and etc. It is not practical to evaluate the design choices one by one.

Based on the above facts, a natural question arises: Is it possible that a network with low inter-block and low intra-block complexity can achieve SOTA
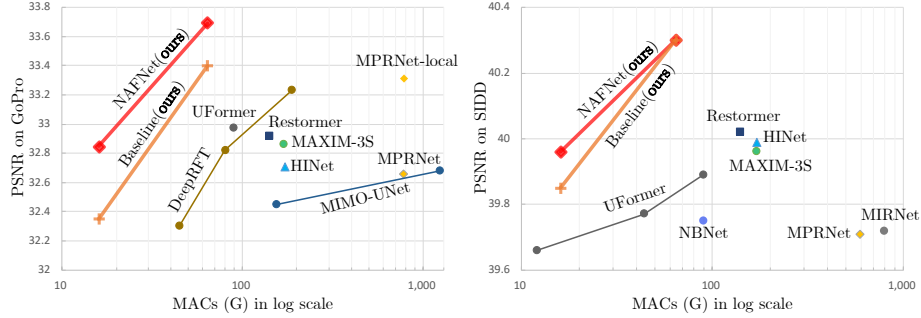

⋆ Equally contribution.

Fig. 1: PSNR vs. computational cost on Image Deblurring (left) and Image Denoising (right) tasks

performance? To accomplish the first condition (low inter-block complexity), this paper adopts the single-stage UNet as architecture (following some SOTA methods[37,34]) and focuses on the second condition. To this end, we start with a plain block with the most common components, i.e. convolution, ReLU, and shortcut[13]. From the plain block, we add/replace components of SOTA methods and verify how much performance gain do these components bring. By extensive ablation studies, we propose a simple baseline, as shown in Figure 3c, that exceeds the SOTA methods and is computationally efficient. It has the potential to inspire new ideas and make their verification easier. The baseline, which contains GELU[14] and Channel Attention Module[15] (CA), can be further simplified: we reveal that the GELU in the baseline can be regarded as a special case of the Gated Linear Unit[9] (GLU), and from this we empirically demonstrate that it can be replaced by a simple gate, i.e. element-wise product of feature maps. In addition, we reveal the similarity of the CA to GLU in form, and the nonlinear activation functions in CA could be removed either. In conclusion, the simple baseline could be further simplified to a nonlinear activation free network, noted as NAFNet. We mainly conduct experiments on SIDD[1] for image denoising, and GoPro[25] for image deblurring, following [4,37,35]. The main results are shown in Figure 1, our proposed baseline and NAFNet achieves SOTA results while being computationally efficient: 33.40/33.69 dB on GoPro, exceed previous SOTA[7] 0.09/0.38 dB, respectively, with 8.4% of its computational cost; 40.30 dB on SIDD, exceed [37] 0.28 dB with less than half of its computational costs. Extensive quantity and quality experiments are conducted to illustrate the effectiveness of our proposed baselines.

The contributions of this paper are summarized as follows:

1. By decomposing the SOTA methods and extracting their essential components, we form a baseline (in Figure 3c) with lower system complexity, which can exceed the previous SOTA methods and has a lower computational cost, as shown in Figure 1. It may facilitate the researchers to inspire new ideas and evaluate them conveniently.

2. By revealing the connections between GELU, Channel Attention to Gated Linear Unit, we further simplify the baseline by removing or replacing the nonlinear activation functions (e.g. Sigmoid, ReLU, and GELU), and propose a nonlinear activation free network, namely NAFNet. It can match or surpass the baseline although being simplified. To the best of our knowledge, it is the first work demonstrates that the nonlinear activation functions may not be necessary for SOTA computer vision methods. This work may have the potential to expand the design space of SOTA computer vision methods.

2   Related Works

2.1   Image Restoration

Image restoration tasks aim to restore a degraded image (e.g. noisy, blur) to a clean one. Recently, deep learning based methods[4,35,37,34,5,6,30,7,24] achieve SOTA results on these tasks, and most of the methods could be viewed as variants of a classical solution, UNet[28]. It stacks blocks to a U-shaped architecture with skip-connection. The variants bring performance gain, as well as the system complexity, and we broadly categorized the complexity as inter-block complexity and intra-block complexity.

Inter-block Complexity [35,4] are multi-stage networks, i.e. the latter stage refine the results of the previous stage, and each stage is a U-shaped architecture. This design is based on the assumption that breaking down the difficult image restoration task into several subtasks contributes to performance. Differently, [6,24] adopt the single-stage design and achieve competitive results, but they introduce complicated connections between various sized feature maps. Some methods adopt the above strategies both, e.g. [30]. Other SOTA methods, e.g. [37,34] maintain the simple structure of single-stage UNet, yet they introduce intra-block complexity, which we will discuss next.

Intra-block Complexity There are numerous different intra-block design schemes, we pick a few examples here. [37] reduces the memory and time complexity of self-attention[32] by channelwise attention map rather than spatialwise. Besides, gated linear units[9] and depthwise convolution are adopted in the feed-forward network. [34] introduces window-based multi-head self-attention, which is similar to [21]. In addition, it introduces locally-enhanced feed-forward network in its block, which adds depthwise convolution to feed-forward network to enhance the local information capture ability. Differently, we reveal that increasing system complexity is not the only way to improve performance: SOTA performance could be achieved by a simple baseline.

2.2   Gated Linear Units

Gated Linear Units[9] (GLU) can be interpreted by the element-wise production of two linear transformation layers, one of which is activated with the nonlinear-

(a) Multi-Stage Architecture

(b) Multi-Scale Fusion Architecture
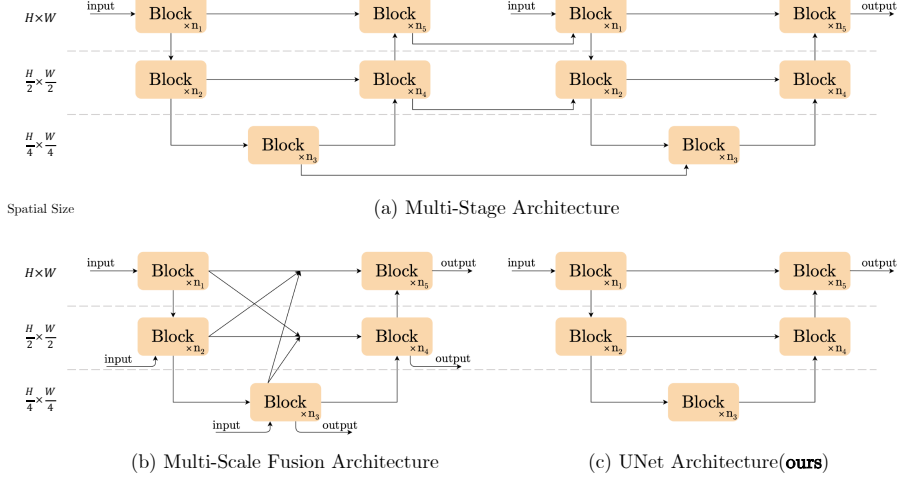
(c) UNet Architecture(**ours**)

Fig. 2: Comparison of architectures of image restoration models. Dashes to distinguish features of different sizes. (a) The multi-stage architecture[4,35] stacks UNet architecture serially. (b) The multi-scale fusion architecture[24,6] fusions the features in different scales. (c)UNet architecture, which is adopted by some SOTA methods[37,34]. We use it as our architecture. Some details have been deliberately omitted for simplicity, e.g. downsample/upsample layers, feature fusion modules, input/output shortcut, and etc.

ity. GLU or its variants has verified their effectiveness in NLP[29,9,8], and there is a prosperous trend of them in computer vision[30,37,16,19]. In this paper, we reveal the non-trivial improvement brought by GLU. Different from [29], we remove the nonlinear activation function in GLU without performance degradation. Furthermore, based on the fact that the nonlinear activation free GLU contains nonlinearity itself (as the product of two linear transformations raises nonlinearity), our baseline could be simplified by replacing the nonlinear activation functions with the multiplication of two feature maps. To the best of our knowledge, it is the first computer vision model achieves SOTA performance without nonlinear activation functions.

3   Build A Simple Baseline

In this section, we build a simple baseline for image restoration tasks from scratch. To keep the structure simple, our principle is not to add entities if they are not necessary. The necessity is verified by empirical evaluation of restoration tasks. We mainly conduct experiments with the model size around 16 GMACs following HINet Simple[4], and the MACs are estimated by an input with the spatial size of 256×256. The results of models with different capacities, e.g. 1.1 GMACs, 65 GMACs, are in the experimental section. We mainly validate the

results (PSNR) on two popular datasets for denoising (i.e. SIDD[1]) and deblurring (i.e. GoPro[25] dataset), based on the fact that those tasks are fundamental in low-level vision. The design choices are discussed in the following subsections.

3.1   Architecture

To reduce the inter-block complexity, we adopt the classic single-stage U-shaped architecture with skip-connections, as shown in Figure 2c, following [37,34]. We believe the architecture will not be a barrier to performance. The experimental results confirmed our conjecture, in Table 6, 7 and Figure 1.

3.2   A Plain Block

Neural Networks are stacked by blocks. We have determined how to stack blocks in the above (i.e. stacked in a UNet architecture), but how to design the internal structure of the block is still a problem. We start from a plain block with the most common components, i.e. convolution, ReLU, and shortcut[13], and the arrangement of these components follows [12,21], as shown in Figure 3b. We will note it as PlainNet for simplicity. Using a convolution network instead of a transformer is based on the following considerations. First, although transformers show good performance in computer vision, some works[12,22] claim that they may not be necessary for achieving SOTA results. Second, depthwise convolution is simpler than the self-attention[32] mechanism. Third, this paper is not intended to discuss the advantages and disadvantages of transformers and convolutional neural networks, but just to provide a simple baseline. The discussion of the attention mechanism is proposed in the subsequent subsection.

3.3   Normalization

Normalization is widely adopted in high-level computer vision tasks, and there is also a popular trend in low-level vision. Although [25] abandoned Batch Normalization[17] as the small batch size may bring the unstable statistics[36], [4] re-introduce the Instance Normalization[31] and avoids the small batch size issue. However, [4] shows that adding instance normalization does not always bring performance gains and requires manual tuning. Differently, under the prosperity of transformers, Layer Normalization[2] is used by more and more methods, including SOTA methods[30,37,34,22,21]. Based on these facts we conjecture Layer Normalization may be crucial to SOTA restorers, thus we add Layer Normalization to the plain block described above. This change can make training smooth, even with a 10× increase in learning rate. The larger learning rate brings significant performance gain: +0.44 dB (39.29 dB to 39.73 dB) on SIDD[1], +3.39 dB (28.51 dB to 31.90 dB) on GoPro[25] dataset. To sum up, we add Layer Normalization to the plain block as it can stabilize the training process.
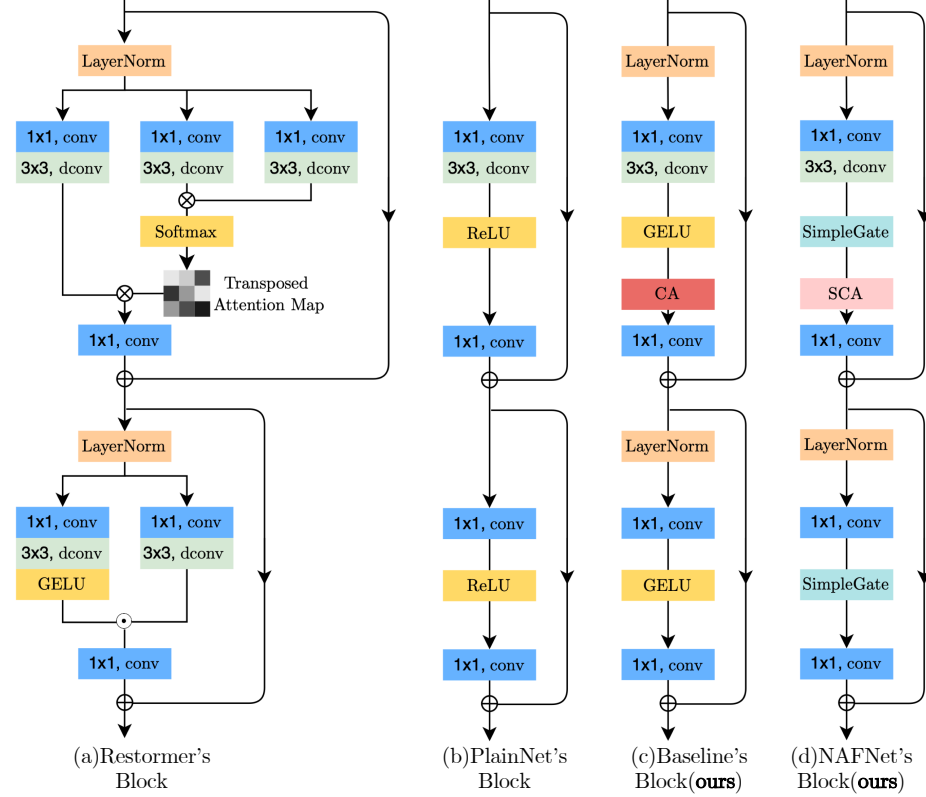
Fig. 3: Intra-block structure comparison. ⊗:matrix multiplication, ⊙/⊕:element-wise multiplication/addition. dconv: Depthwise convolution. Nonlinear activation functions are represented by yellow boxes. (a) Restormer's block[37], some details are omitted for simplicity, e.g. reshaping the feature maps. (b) PlainNet's block, which contains the most common components. (c) Our proposed baseline. Compares to (b), Channel Attention (CA) and LayerNorm are adopted. Besides, ReLU is replaced by GELU. (d) Our proposed Nonlinear Activation Free Network's block. It replaces CA/GELU with Simplified Channel Attention(SCA) and SimpleGate respectively. The details of these components are shown in Fig 4

3.4   Activation

The activation function in the plain block, Rectified Linear Unit[27] (ReLU), is extensively used in computer vision. However, there is a tendency to replace ReLU with GELU[14] in SOTA methods[22,37,30,21,11]. This replacement is implemented in our model either. The performance stays comparable on SIDD (from 39.73 dB to 39.71 dB) which is consistent with the conclusion of [22], yet it brings 0.21 dB performance gain (31.90 dB to 32.11 dB) on GoPro. In short, we

replace ReLU with GELU in the plain block, because it keeps the performance of image denoising while bringing non-trivial gain on image deblurring.

3.5   Attention

Considering the recent popularity of the transformer in computer vision, its attention mechanism is an unavoidable topic in the design of the internal structure of the block. There are many variants of attention mechanisms, and we discuss only a few of them here. The vanilla self-attention mechanism[32], which is adopted by [11,3], generate the target feature by the linear combination of all features which are weighted by the similarity between them. Therefore, each feature contains global information, while it suffers from the quadratic computational complexity with the size of the feature map. Some image restoration tasks process data at high resolution which makes the vanilla self-attention not practical. Alternatively, [21,20,34] apply self-attention only in a fix-sized local window to alleviate the issue of increased computation. While it lacks global information. We do not take the window-based attention, as the local information could be well captured by the depthwise convolution [12,22] in the plain block.

Differently, [37] modifies the spatial-wise attention to channel-wise, avoids the computation issue while maintaining global information in each feature. It could be seen as a special variant of channel attention [15]. Inspired by [37], we realize the vanilla channel attention meets the requirements: computational efficiency and brings global information to the feature map. In addition, the effectiveness of channel attention has been verified in the image restoration task[35,7], thus we add the channel attention to the plain block. It brings 0.14 dB on SIDD[1] (39.71 dB to 39.85 dB), 0.24 dB on GoPro[25] dataset (32.11 dB to 32.35 dB).

3.6   Summary

So far, we build a simple baseline from scratch, as we shown in Table 1. The architecture and the block are shown in Figure 2c and Figure 3c, respectively. Each component in the baseline is trivial, e.g. Layer Normalization, Convolution, GELU, and Channel Attention. But the combination of these trivial components leads to a strong baseline: it can surpass the previous SOTA results on SIDD and GoPro dataset with only a fraction of computation costs, as we shown in Figure 1 and Table 6,7. We believe the simple baseline could facilitate the researchers to evaluate their ideas.

4   Nonlinear Activation Free Network

The baseline described above is simple and competitive, but is it possible to further improve performance while ensuring simplicity? Can it be simpler without performance loss? We try to answer these questions by looking for commonalities from some SOTA methods[30,37,19,16]. We find that in these methods, Gated Linear Units[9](GLU) are adopted. It implies that GLU might be promising. We will discuss it next.
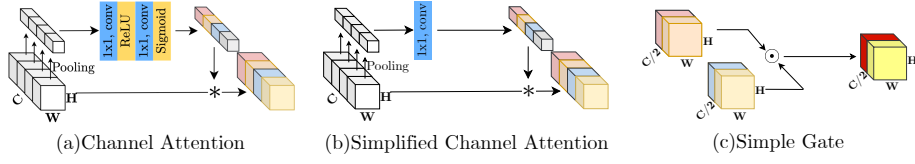
(a)Channel Attention   (b)Simplified Channel Attention   (c)Simple Gate

Fig. 4: Illustration of (a) Channel Attention[15] (CA), (b) Simplified Channel Attention (SCA), and (c) Simple Gate (SG). $\odot/*$: element-wise/channel-wise multiplication

Gated Linear Units The gated linear units could be formulated as:

$$Gate(\mathbf{X}, f, g, \sigma) = f(\mathbf{X}) \odot \sigma(g(\mathbf{X})), \tag{1}$$

where $\mathbf{X}$ represents the feature map, f and g are linear transformers, σ is a non-linear activation function, e.g. Sigmoid, and $\odot$ indicates element-wise multiplication. As discussed above, adding GLU to our baseline may improve the performance yet the intra-block complexity is increasing as well. This is not what we expected. To address this, we revisit the activation function in the baseline, i.e. GELU[14]:

$$GELU(x) = x\Phi(x), \tag{2}$$

where Φ indicates the cumulative distribution function of the standard normal distribution. And based on [14], GELU could be approximated and implemented by:

$$0.5x(1 + tanh[\sqrt{2/\pi}(x + 0.044715x^3)]). \tag{3}$$

From Eqn. 1 and Eqn. 2, it can be noticed that GELU is a special case of GLU, i.e. f, g are identity functions and take σ as Φ. Through the similarity, we conjecture from another perspective that GLU may be regarded as a generalization of activation functions, and it might be able to replace the nonlinear activation functions. Further, we note that the GLU itself contains nonlinearity and does *not* depend on σ: even if the σ is removed, $Gate(\mathbf{X}) = f(\mathbf{X}) \odot g(\mathbf{X})$ contains nonlinearity. Based on these, we propose a simple GLU variant: directly divide the feature map into two parts in the channel dimension and multiply them, as we shown in Figure 4c, noted as SimpleGate. Compared to the complicated implementation of GELU in Eqn.3, our SimpleGate could be implemented by an element-wise multiplication, that's all:

$$SimpleGate(\mathbf{X}, \mathbf{Y}) = \mathbf{X} \odot \mathbf{Y}, \tag{4}$$

where $\mathbf{X}$ and $\mathbf{Y}$ are feature maps of the same size.

By replacing GELU in the baseline to the proposed SimpleGate, the performance of image denoising (on SIDD[1]) and image deblurring (on GoPro[25] dataset) boost 0.08 dB (39.85 dB to 39.93 dB) and 0.41 dB (32.35 dB to 32.76 dB) respectively. The results demonstrate that GELU could be replaced by our proposed SimpleGate. At this point, only a few types of nonlinear activations left in the network: Sigmoid and ReLU in the channel attention module[15], and we will discuss the simplifications of it next.

Simplified Channel Attention In Section 3, we adopt the channel attention[15] into our block as it captures the global information and it is computationally efficient. It is illustrated in Figure 4a: it squeezes the spatial information into channels first and then a multilayer perceptual applies to it to calculate the channel attention, which will be used to weight the feature map. It could be represented as:

$$CA(\mathbf{X}) = \mathbf{X} * \sigma(W_2 max(0, W_1 pool(\mathbf{X}))),\qquad(5)$$

where $\mathbf{X}$ represents the feature map, *pool* indicates the global average pooling operation which aggregates the spatial information into channels. σ is a nonlinear activation function, Sigmoid, W_1, W_2 are fully-connected layers and ReLU is adopted between two fully-connected layers. Last, $*$ is a channelwise product operation. If we regard the channel-attention calculation as a function, noted as Ψ with input $\mathbf{X}$, Eqn. 5 could be re-written as:

$$CA(\mathbf{X}) = \mathbf{X} * \Psi(\mathbf{X}).\qquad(6)$$

It can be noticed that Eqn. 6 is very similar to Eqn. 1. This inspires us to consider channel attention as a special case of GLU, which can be simplified like GLU in the previous subsection. By retaining the two most important roles of channel attention, that is, aggregating global information and channel information interaction, we propose the Simplified Channel Attention:

$$SCA(\mathbf{X}) = \mathbf{X} * W pool(\mathbf{X}).\qquad(7)$$

The notations follows Eqn. 5. Apparently, Simplified Channel Attention (Eqn. 7) is simpler than the original one (Eqn. 5), as shown in Figure 4a and Figure 4b. Although it is simpler, there is no loss of performance: +0.03 dB (39.93 dB to 39.96 dB) on SIDD and +0.09 dB (32.76 dB to 32.85 dB) on GoPro.

Summary Starting from the baseline proposed in Section 3, we further simplify it by replacing the GELU with SimpleGate and Channel Attention to Simplified Channel Attention, without loss of performance. We emphasize that after the simplification, there are *no* nonlinear activation functions (e.g. ReLU, GELU, Sigmoid, etc.) in the network. So we call this baseline Nonlinear Activation Free Network, namely NAFNet. It can match or surpass the baseline although without nonlinear activation functions, as we shown in Figure 1 and Table 6,7. We can now answer the questions in the begining of this section by yes, because of the simplicity and effectiveness of NAFNet.

5   Experiments

In this section, we analyze the effect of the design choices of NAFNet described in previous sections in detail. Next, we apply our proposed NAFNet to various image restoration applications, including RGB image denoising, image deblurring, raw image denoising, and image deblurring with JPEG artifacts.
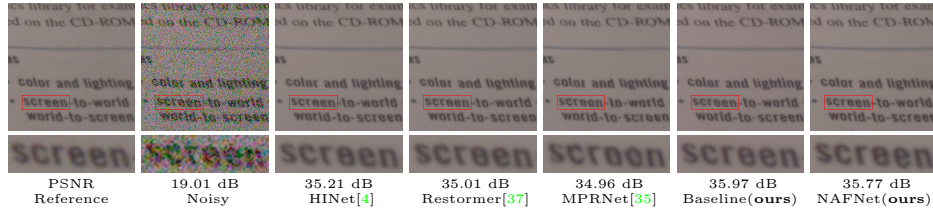
| PSNR
Reference | 19.01 dB
Noisy | 35.21 dB
HINet[4] | 35.01 dB
Restormer[37] | 34.96 dB
MPRNet[35] | 35.97 dB
Baseline(**ours**) | 35.77 dB
NAFNet(**ours**) |

Fig. 5: Qualitative comparison of image denoising methods on SIDD[1]

5.1 Ablations

The ablation studys are conducted on image denoising (SIDD[1]) and deblurring (GoPro[25]) tasks. We follow experiments setting of [4] if not specified, e.g. 16 GMACs of computational budget, gradient clip, and PSNR loss. We train models with Adam[18] optimizer ($\beta_1 = 0.9, \beta_2 = 0.9$, weight decay 0) for total 200K iterations with the initial learning rate $1e^{-3}$ gradually reduced to $1e^{-6}$ with the cosine annealing schedule[23]. The training patch size is 256×256 and batch size is 32. Training by patches and testing by the full image raises performance degradation[7], we solve it by adopting TLC[7] following MPRNet-local[7]. The effectiveness of TLC on GoPro[1] is shown in Tab 4. We mainly compare TLC with "test by patches" strategy, which is adopted by [4], [24], and etc. It brings performance gains and avoids the artifacts brought by patches. Moreover, we apply skip-init[10] to stabilize training following [22]. The default width and number of blocks are 32 and 36, respectively. We adjust the width to keep the computational budget hold if the number of blocks changed. We report Peak Signal to Noise Ratio (PSNR) and Structural SIMilarity (SSIM) in our experiments. The speed/memory/computational complexity evaluation is conducted with an input size of 256×256, on an NVIDIA 2080Ti GPU.

From PlainNet to the simple baseline: PlainNet is defined in Section 3, and its block is illustrated in Figure 3b. We find that the training of PlainNet is unstable under the default settings. As an alternative, we reduce the learning rate (lr) by a factor of 10 to make the model trainable. This issue is solved by introducing Layer Normalization (LN): the learning rate can be increased from $1e^{-4}$ to $1e^{-3}$ with a more stable training process. In PSNR, LN brings 0.46 dB and 3.39 dB on SIDD and GoPro respectively. Besides, GELU and Channel Attention (CA) also demonstrated their effectiveness in Table 1.

From the simple baseline to NAFNet: As described in Section 3, NAFNet can be obtained by simplifying the baseline. In Tab 2, we show that there is no performance penalty for this simplification. Instead, the PSNR boosts 0.11 dB and 0.50 dB in SIDD and GoPro respectively. The computational complexity

---

[1] SIDD test on 256×256 patches avoid the inconsistent issue.

Table 1: Build a simple baseline from PlainNet. The effectiveness of Layer Normalization (LN), GELU, and Channel Attention (CA) have been verified. ∗ indicates that the training is unstable due to the large learning rate (lr)

	lr	LN	ReLU→GELU	CA	SIDD PSNR SSIM		GoPro PSNR SSIM	
PlainNet	$1e^{-4}$				39.29	0.956	28.51	0.907
PlainNet*	$1e^{-3}$				-	-	-	-
	$1e^{-3}$	✓			39.73	0.959	31.90	0.952
	$1e^{-3}$	✓	✓		39.71	0.958	32.11	0.954
Baseline	$1e^{-3}$	✓	✓	✓	39.85	0.959	32.35	0.956

Table 2: NAFNet is derived from the simplification of baseline, i.e. replacing GELU to SimpleGate (SG), and replacing Channel Attention (CA) to Simplified Channel Attention (SCA).

	GELU→SG	CA→SCA	SIDD PSNR SSIM		GoPro PSNR SSIM		speedup
Baseline			39.85	0.959	32.35	0.956	1.00×
	✓		39.93	0.960	32.76	0.960	0.98×
		✓	39.95	0.960	32.54	0.958	1.11×
NAFNet	✓	✓	39.96	0.960	32.85	0.960	1.09×

is consistent for a fair comparison, and details in the supplementary material. The speedup of modifications compared to Baseline is provided. In addition, no significant extra memory consumption compares to Baseline in inference.

Number of blocks: We verify the effect of the number of blocks on NAFNet in Table 3. We mainly consider the latency at spatial size 720×1280, as this is the size of the entire GoPro image. In the process of increasing the number of blocks to 36, the performance of the model has been greatly improved, and the latency has not increased significantly (+14.5% compares to 9 blocks). When the number of blocks further increases to 72, the performance improvement of the model is not obvious, but the latency increases significantly (+30.0% compares to 36 blocks). Because 36 blocks can achieve a better performance/latency balance, we use it as the default option.

Variants of σ in SimpleGate: Vanilla gated linear unit (GLU) contains a nonlinear activation function σ as formulated in Eqn. 1. Our proposed SimpleGate, as shown in Eqn. 4 and Figure 4c removes it. In other words, σ in SimpleGate is set as an identity function. We variants the σ from the identity function to different nonlinear activation functions in Table 5 to judge the importance of nonlinearity in σ. PSNR on SIDD is basically unaffected (fluctuates from 39.96

Table 3: The effect of the number of blocks. The width is adjusted to keep the computational budget hold. Latency-256 and Latency-720 is based on the input size 256×256 and 720×1280 respectively, in milliseconds

	# of blocks	SIDD PSNR SSIM	GoPro PSNR SSIM	Latency-256	Latency-720
NAFNet	9	39.78 0.959	31.79 0.951	11.8	154.7
	18	39.90 0.960	32.64 0.951	19.9	151.7
	36	39.96 0.960	32.85 0.959	39.1	177.1
	72	39.95 0.960	32.88 0.961	73.8	230.1

Table 4: Effectiveness of TLC[7] on GoPro[25]

	patches?	TLC?	PSNR	SSIM
			33.08	0.963
NAFNet	✓		33.65	0.966
		✓	33.69	0.967

Table 5: Variants of σ in $SimpleGate(\mathbf{X}, \mathbf{Y}) = \mathbf{X} \odot \sigma(\mathbf{Y})$

σ	SIDD PSNR	SSIM	GoPro PSNR	SSIM
Identity(**ours**)	39.96	0.960	32.85	0.960
ReLU	39.98	0.960	32.59	0.958
GELU	39.97	0.960	32.72	0.959
Sigmoid	39.99	0.960	32.50	0.958
SiLU	39.96	0.960	32.74	0.960

dB to 39.99 dB), while PSNR on GoPro drops significantly (-0.11 dB to -0.35 dB), which indicates that in NAFNet, the σ in SimpleGate may not be necessary.

5.2   Applications

We apply NAFNet to various image restoration tasks, follow the training settings of ablation study if not specified, except that it is enlarged by increasing the width from 32 to 64. Besides, batch size and total training iterations are 64 and 400K respectively, following [4]. Random crop augmentation is applied. We report the mean of three experimental results.

RGB Image Denoising We compare the RGB Image Denoising results with other SOTA methods on SIDD, show in Table 6. Baseline and its simplified version NAFNet, exceed the previous best result Restormer 0.28 dB with only a fraction of its computational cost, as shown in Figure 1. The qualitative results are shown in Figure 5. Our proposed baselines can restore more fine details compared to other methods. Moreover, we achieve SOTA result (40.15 dB) on the online benchmark, exceed previous top-ranked methods 0.23 dB.

Image Deblurring We compare the deblurring results of SOTA methods on GoPro[25] dataset, flip and rotate augmentations are adopted. As we shown in Table 7 and Figure 1, our baseline and NAFNet surpass the previous best method MPRNet-local[7] 0.09 dB and 0.38 dB in PSNR, respectively, with only 8.4% of its computational costs. The visualization results are shown in Figure 6, our baselines can restore sharper results compares to other methods.

23.21 dB	23.21	29.68 dB	31.58 dB
Blurry Image	Blurry	Restormer [37]	Baseline(**ours**)
Reference	28.89 dB	30.35 dB	32.54 dB
PSNR	MPRNet [35]	MPRNet-local [7]	NAFNet(**ours**)

Fig. 6: Qualitative comparison of image deblurring methods on GoPro[25]

Table 6: Image Denoising Results on SIDD[1]

Method	MPRNet [35]	MIRNet [38]	NBNet [5]	UFormer [34]	MAXIM [30]	HINet [4]	Restormer [37]	Baseline **ours**	NAFNet **ours**
PSNR	39.71	39.72	39.75	39.89	39.96	39.99	40.02	40.30	40.30
SSIM	0.958	0.959	0.959	0.960	0.960	0.958	0.960	0.962	0.962
MACs(G)	588	786	88.8	89.5	169.5	170.7	140	65	65

Raw Image Denoising We apply NAFNet to a raw image denoising task. The training and testing settings follow PMRID[33], and we noted the testing set as 4Scenes (as the dataset contains 39 raw images of 4 different scenes in various light conditions) for simplicity. In addition, we make fair comparison by changing the width and number of blocks of NAFNet from 32 to 16, 36 to 7, respectively, so that the computational cost is less than PMRID. The results shown in Table 8 and Figure 7 demonstrate NAFNet can surpass PMRID quantitatively and qualitatively. In addition, this experiment indicates our NAFNet can be scaled flexibly (from 1.1 GMACs to 65 GMACs).

Image Deblurring with JPEG artifacts We conduct experiments on REDS[26] dataset, the training setting follows [4,30], and we evaluate the result on 300 images from the validation set (noted as REDS-val-300) following [4,30]. As shown in Table 9, our method outperforms other competing methods, including the

Table 7: Image Deblurring Results on GoPro[25]

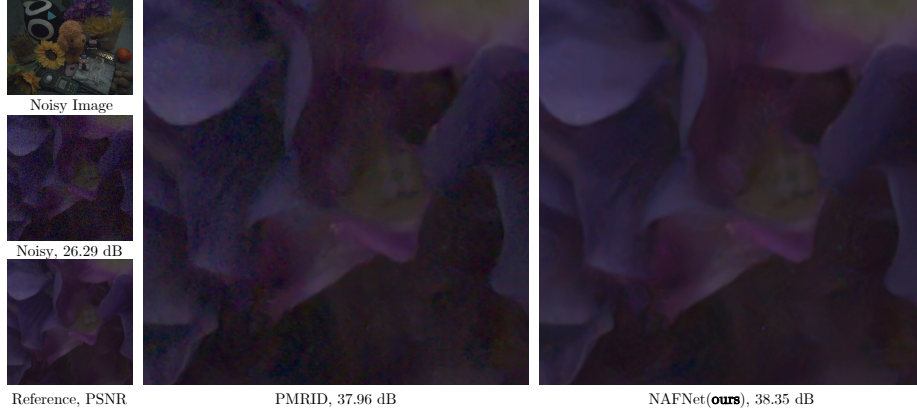| Method	MIMO-UNet [6]	HINet [4]	MAXIM [30]	Restormer [37]	UFormer [34]	DeepRFT [24]	MPRNet -local[7]	Baseline **ours**	NAFNet **ours**
PSNR	32.68	32.71	32.86	32.92	32.97	33.23	33.31	33.40	33.69
SSIM	0.959	0.959	0.961	0.961	0.967	0.963	0.964	0.965	0.967
MACs(G)	1235	170.7	169.5	140	89.5	187	778.2	65	65

Fig. 7: Qualitatively compare the noise reduction effects of PMRID[33] and our porposed NAFNet. Zoom in to see details

Table 8: Raw image denoising results on 4Scenes[33]

Method	PSNR	SSIM	MACs(G)
PMRID[33]	39.76	0.975	1.2
NAFNet(**ours**)	40.05	0.977	1.1

Table 9: Image deblurring results on REDS-val-300[26]

Method	PSNR	SSIM	MACs(G)
MPRNet[35]	28.79	0.811	776.7
HINet[4]	28.83	0.862	170.7
MAXIM[30]	28.93	0.865	169.5
NAFNet(**ours**)	29.09	0.867	65

previous winning solution (HINet) on the REDS dataset of NTIRE 2021 Image Deblurring Challenge Track2 JPEG artifacts[26].

6   Conclusions

By decomposing the SOTA methods, we extract the essential components and adopt them on a naive PlainNet. The obtained baseline reaches SOTA performance on image denoising and image deblurring tasks. By analyzing the baseline, we reveal that it can be further simplified: The nonlinear activation functions in it can be completely replaced or removed. From this, we propose a nonlinear activation free network, NAFNet. Although simplified, its performance is equal to or better than baseline. Our proposed baselines may facilitate the researchers to evaluate their ideas. In addition, this work has the potential to influence future computer vision model design, as we demonstrate that nonlinear activation functions are not necessary to achieve SOTA performance.

References

1. Abdelhamed, A., Lin, S., Brown, M.S.: A high-quality denoising dataset for smartphone cameras. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
2. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
3. Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., Gao, W.: Pre-trained image processing transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12299–12310 (2021)
4. Chen, L., Lu, X., Zhang, J., Chu, X., Chen, C.: Hinet: Half instance normalization network for image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 182–192 (2021)
5. Cheng, S., Wang, Y., Huang, H., Liu, D., Fan, H., Liu, S.: Nbnet: Noise basis learning for image denoising with subspace projection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4896–4906 (2021)
6. Cho, S.J., Ji, S.W., Hong, J.P., Jung, S.W., Ko, S.J.: Rethinking coarse-to-fine approach in single image deblurring. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4641–4650 (2021)
7. Chu, X., Chen, L., , Chen, C., Lu, X.: Improving image restoration by revisiting global information aggregation. arXiv preprint arXiv:2112.04491 (2021)
8. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q.V., Salakhutdinov, R.: Transformer-xl: Attentive language models beyond a fixed-length context. arXiv preprint arXiv:1901.02860 (2019)
9. Dauphin, Y.N., Fan, A., Auli, M., Grangier, D.: Language modeling with gated convolutional networks. In: International conference on machine learning. pp. 933–941. PMLR (2017)
10. De, S., Smith, S.: Batch normalization biases residual blocks towards the identity function in deep networks. Advances in Neural Information Processing Systems **33**, 19964–19975 (2020)
11. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
12. Han, Q., Fan, Z., Dai, Q., Sun, L., Cheng, M.M., Liu, J., Wang, J.: Demystifying local vision transformer: Sparse connectivity, weight sharing, and dynamic weight. arXiv preprint arXiv:2106.04263 (2021)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
14. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 (2016)
15. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141 (2018)
16. Hua, W., Dai, Z., Liu, H., Le, Q.V.: Transformer quality in linear time. arXiv preprint arXiv:2202.10447 (2022)
17. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. PMLR (2015)

18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
19. Liang, J., Cao, J., Fan, Y., Zhang, K., Ranjan, R., Li, Y., Timofte, R., Van Gool, L.: Vrt: A video restoration transformer. arXiv preprint arXiv:2201.12288 (2022)
20. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., Timofte, R.: Swinir: Image restoration using swin transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1833–1844 (2021)
21. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022 (2021)
22. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. arXiv preprint arXiv:2201.03545 (2022)
23. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016)
24. Mao, X., Liu, Y., Shen, W., Li, Q., Wang, Y.: Deep residual fourier transformation for single image deblurring. arXiv preprint arXiv:2111.11745 (2021)
25. Nah, S., Hyun Kim, T., Mu Lee, K.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3883–3891 (2017)
26. Nah, S., Son, S., Lee, S., Timofte, R., Lee, K.M.: Ntire 2021 challenge on image deblurring. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 149–165 (2021)
27. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Icml (2010)
28. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
29. Shazeer, N.: Glu variants improve transformer. arXiv preprint arXiv:2002.05202 (2020)
30. Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A., Li, Y.: Maxim: Multi-axis mlp for image processing. arXiv preprint arXiv:2201.02973 (2022)
31. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)
32. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
33. Wang, Y., Huang, H., Xu, Q., Liu, J., Liu, Y., Wang, J.: Practical deep raw image denoising on mobile devices. In: European Conference on Computer Vision. pp. 1–16. Springer (2020)
34. Wang, Z., Cun, X., Bao, J., Liu, J.: Uformer: A general u-shaped transformer for image restoration. arXiv preprint arXiv:2106.03106 (2021)
35. Waqas Zamir, S., Arora, A., Khan, S., Hayat, M., Shahbaz Khan, F., Yang, M.H., Shao, L.: Multi-stage progressive image restoration. arXiv e-prints pp. arXiv–2102 (2021)
36. Yan, J., Wan, R., Zhang, X., Zhang, W., Wei, Y., Sun, J.: Towards stabilizing batch statistics in backward propagation of batch normalization. arXiv preprint arXiv:2001.06838 (2020)
37. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H.: Restormer: Efficient transformer for high-resolution image restoration. arXiv preprint arXiv:2111.09881 (2021)

38. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Learning enriched features for real image restoration and enhancement. In: European Conference on Computer Vision. pp. 492–511. Springer (2020)