



Efficient Cascaded Multiscale Adaptive Network for Image Restoration

Yichen Zhou^{1,2} , Pan Zhou³ , and Teck Khim Ng¹

¹ National University of Singapore

² Sea AI Lab

³ Singapore Management University

zhou.yichen@u.nus.edu panzhou@smu.edu.sg ngtk@comp.nus.edu.sg

Abstract. Image restoration, encompassing tasks such as deblurring, denoising, and super-resolution, remains a pivotal area in computer vision. However, efficiently addressing the spatially varying artifacts of various low-quality images with local adaptiveness and handling their degradations at different scales poses significant challenges. To efficiently tackle these issues, we propose the novel *Efficient Cascaded Multiscale Adaptive* (ECMA) Network. ECMA employs Local Adaptive Module, LAM, which dynamically adjusts convolution kernels across local image regions to efficiently handle varying artifacts. Thus, LAM addresses the local adaptiveness challenge more efficiently than costlier mechanisms like self-attention, due to its less computationally intensive convolutions. To construct a basic ECMA block, three cascading LAMs with convolution kernels from large to small sizes are employed to capture features at different scales. This cascaded multiscale learning effectively handles degradations at different scales, critical for diverse image restoration tasks. Finally, ECMA blocks are stacked in a U-Net architecture to build ECMA networks, which efficiently achieve both local adaptiveness and multiscale processing. Experiments show ECMA’s high performance and efficiency, achieving comparable or superior restoration performance to state-of-the-art methods while reducing computational costs by $1.2\times$ to $9.7\times$ across various image restoration tasks, e.g., image deblurring, denoising and super-resolution.

Keywords: Image Deblurring · Image Denoising · Image Super-resolution, Image Restoration

1 Introduction

Image restoration is a pivotal task within computer vision, encompassing a diverse array of subtasks with numerous applications [28, 33, 37]. These subtasks include deblurring [15, 37–39, 58], denoising [1, 28], and super-resolution [3, 16, 35, 36, 56], each playing a significant role in addressing real-world challenges. The central objective across these subtasks is to recover a high-quality image from an initial, degraded, and low-quality input, with implications across various domains and applications [1, 3, 33, 37, 39].

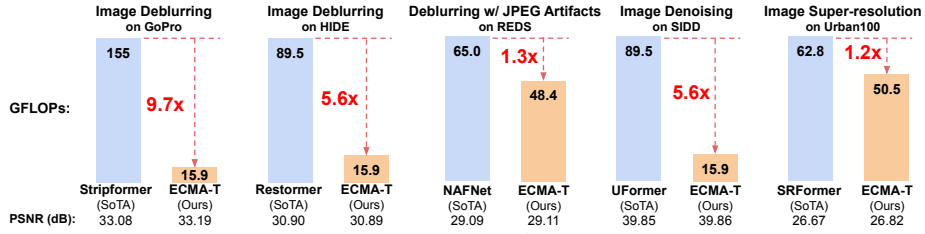


Fig. 1: Computation cost comparison of ECMA and corresponding state-of-the-art methods of similar performance on various image restoration tasks.

Convolutional Neural Networks (CNNs) have emerged as prominent architectures for image restoration, demonstrating impressive performance through hierarchical feature extraction [19, 45]. However, they exhibit certain limitations, notably in local adaptiveness, multiscale processing, and computational efficiency. Specifically, CNNs [20, 57, 60] use a single uniform convolution kernel across all spatial regions in each layer, assuming spatial stationarity. This assumption restricts the ability to effectively handle realistic artifacts that exhibit spatial non-stationarity. Moreover, CNNs primarily focus on local feature processing at a small scale. To gather pivotal contextual information from extensive neighborhood regions for image restoration, CNNs often necessitate large convolution kernels or deeper architectures, increasing computational costs. The advent of transformer networks, such as SwinIR [25], Restormer [53], and GRL [24], has alleviated some of the local adaptiveness challenges through their attention modules [13, 29], since features at different locations are aggregated differently depending on their attention score with respect to other features. However, transformers have not fully addressed issues of fine-grained feature processing and efficiency. While transformers are capable of building long-range dependencies to learn global coarse-grained features, they are less effective in capturing local, detailed features [13, 51, 61]. Accordingly, they do not inherently consider varying image degradation scales, thus do not effectively perform multiscale feature restoration, especially fine-grained features. The computation efficiency is hindered by the quadratic computational complexity of the self-attention modules in transformers. Thus, the following issues remain challenging: how to efficiently achieve both local adaptiveness and multiscale feature processing in image restoration.

To tackle this challenge, in this work, we introduce the Efficient Cascaded Multiscale Adaptive (ECMA) network. ECMA first introduces an efficient Local Adaptive Module, “LAM”, whose convolution kernels can adapt across different local image regions to handle varying degradations effectively. LAM efficiently implements its local adaptive convolution by first using a standard convolution to learn context from extensive neighboring regions, then using this context to guide another convolution for adaptively processing local image regions. This spatial adaptiveness effectively addresses variations in degradations across different regions in an image. Moreover, compared with expensive mechanisms like self-attention, LAM achieves local adaptiveness more efficiently, since convolutions are often more computationally cheap. Next, to build an ECMA block,

ECMA cascades three LAMs whose local adaptive convolution size ranges from large to small for capturing features at different scales. This cascaded multiscale learning allows ECMA to handle degradations at different scales, crucial for tasks like motion blurring which requires large-scale processing, and non-uniform noise or super-resolution that benefit from local processing. Finally, ECMA blocks are stacked like a U-Net architecture to build the ECMA network. Thus, ECMA efficiently achieves both local adaptiveness and multiscale processing through the proposed LAM and cascaded multiscale learning.

Experimental results validate the superior performance and computational efficiency of the proposed ECMA across various image restoration tasks, including deblurring, denoising, and super-resolution. As demonstrated in Figure 1, ECMA achieves higher or comparable image restoration performance with state-of-the-art methods while reducing their computational cost by $1.2\times$ to $9.7\times$. ECMA’s balance of high performance and low computational cost marks a significant advancement in the image restoration tasks.

2 Related Works

Image restoration encompasses deblurring, denoising, and super-resolution etc., each presenting unique challenges. Traditional methods often rely on mathematical modeling and optimization techniques, e.g., variational regularization and Wiener deconvolution [21, 41]. While effective in constrained scenarios, these techniques struggle with complex and spatially varying degradation.

The advent of deep learning has significantly impacted image restoration, with Convolutional Neural Networks (CNNs) becoming the cornerstone. In deblurring, SRN [45] employs recurrent neural networks for iterative refinement, and DeblurGAN [19] utilizes GANs to enhance deblurred image quality. For denoising, DnCNN [46] introduces a deep CNN architecture learning a residual mapping to remove noise. The representative BM3D [5] uses 3D filtering to effectively remove noise while preserving image details. Regarding super-resolution, SRCNN [12] is the first one to apply deep learning to this task. It uses a three-layer CNN to learn end-to-end mapping from low-resolution to high-resolution images. However, CNNs uniformly apply convolution operations across spatial regions, potentially limiting effectiveness against spatially varying artifacts. Additionally, convolution by default only operates on a small local neighbor region, which limits its power in processing features at a larger scale. The integration of contextual information often requires the use of either large convolutional kernels or deep network architectures with many layers, both of which come at a significant computational cost.

Non-local learning mechanisms have been used to improve image restoration performance, e.g., the non-local means for deep denoising [14] and self-attention for super-resolution [60]. Later, transformer models, like Restormer [53] and GRL [24], have employed innovative self-attention to efficiently handle a variety of restoration tasks. But transformers neither explicitly nor efficiently address the varying scales of image degradation, potentially overlooking fine details. Fur-

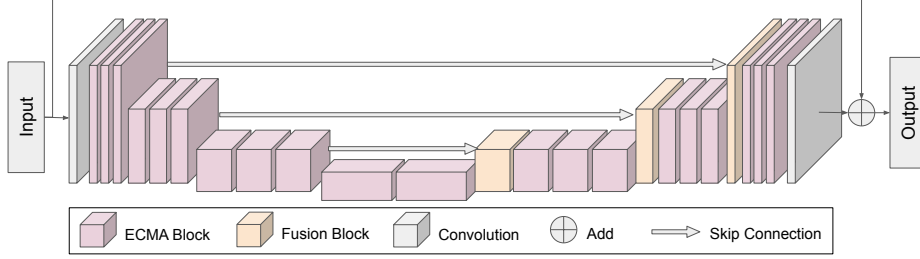


Fig. 2: Overall high-level architecture ECMA networks.

thermore, self-attention introduces quadratic computational complexity relative to image size, thereby limiting their practical efficiency.

3 Methodology

This section elaborates on the detailed structure of Efficient Cascaded Multiscale Adaptive (ECMA) network for image restoration tasks.

3.1 Overview Architecture

As illustrated in Figure 2, ECMA follows other works [48, 51] to adopt a U-Net [40] architecture, integrating a specially designed ECMA block. This efficiently addresses the two challenges: local adaptiveness and multiscale feature processing. Specifically, like most U-Net architectures, ECMA consists of an encoder and a decoder, each comprising several stages. In each encoder stage, ECMA halves the spatial size of feature maps through downsampling and doubles the channel number. In the decoder, each stage doubles the spatial size using transposed convolution and concatenates it with corresponding encoder features, halving the channel number. This hierarchical structure is effective not only for image restoration tasks but also for other applications, such as segmentation.

The main components in ECMA are ECMA blocks and fusion blocks. ECMA block, in particular, processes image features while achieving local adaptiveness and multiscale processing within each block. The fusion block aligns and fuses upsampled decoder features with corresponding encoder features for further processing. In the following, we will introduce these two key blocks in turn.

3.2 ECMA Block

The ECMA block is designed to enhance both local adaptiveness and multiscale processing in image restoration. As shown in Figure 3a, ECMA block employs a Local Adaptive Module (LAM) that uses local adaptive convolution to dynamically process local image features. For multiscale feature processing, the ECMA block incorporates a Cascaded Multiscale Learning (CML) approach. CML cascades three LAMs with varying convolution sizes, from large to small, to process features at different scales. Next, we introduce LAM and CML in detail.

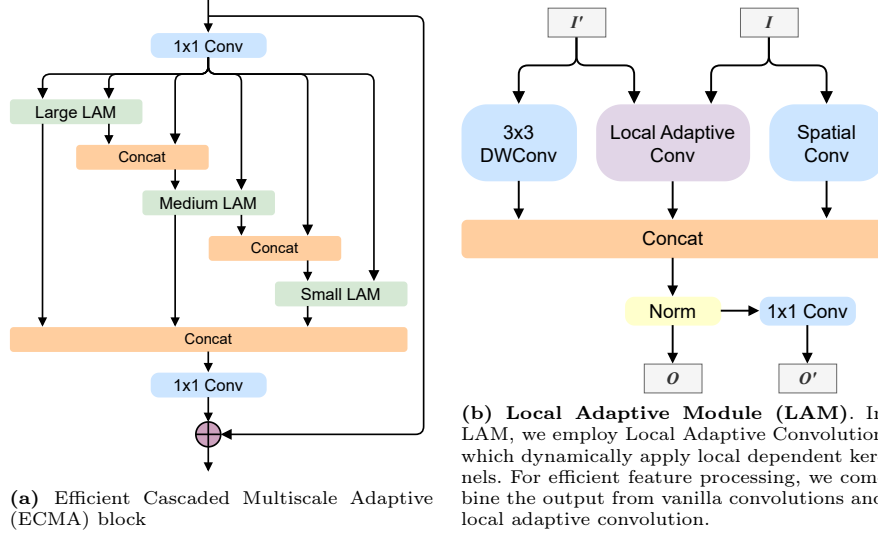


Fig. 3: Structure of ECMA and LAM.

Local Adaptive Module As shown in Figure 3b, the primary component of the Local Adaptive Module (LAM) is a local adaptive convolution, crucial for local adaptive feature processing. Each LAM possesses two inputs, I and I' , where I' denotes the spatial context of I , which is further explained below.

Spatial Context Construction. For any pixel $I'_{i,j}$ computed by using the neighboring features around $I_{i,j}$ in the input of I , then we say I' denotes spatial context of input I . It indeed means that there is spatial correspondence between I and I' . To achieve this, in an ECMA block, I' and I in the first LAM are two splits of the same convolution feature along the channel dimension, thereby having spatial correspondence. For the second and third LAMs, I' denotes the output of the preceding LAM, maintaining spatial correspondence through convolution operations. In this way, each pixel $I'_{i,j}$ in the context I' contains the overall neighboring spatial information around the pixel $I_{i,j}$ in the input I , and can thus be used for local adaptive convolution.

Local Adaptive Convolution. Formally, given the input I and its spatial context I' , our local adaptive convolution, denoted as $\text{Conv}_{\mathbf{W}, \mathbf{W}'}$, is parameterized by two kernels: $\mathbf{W} \in \mathbb{R}^{m \times m}$ and $\mathbf{W}' \in \mathbb{R}^{n \times n}$, and is defined as

$$\mathbf{F} = \text{Conv}_{\mathbf{W}, \mathbf{W}'}(\mathbf{I}, \mathbf{I}'). \quad (1)$$

Here each pixel $F_{s,t}$ in the output \mathbf{F} is computed as

$$F_{s,t} = \sum_{i=1}^m \sum_{j=1}^m \bar{\mathbf{W}}_{i,j} \mathbf{I}_{s-i, t-j}, \quad (2)$$

where $\bar{\mathbf{W}}_{i,j} = \sigma_{s,t} \mathbf{W}_{i,j}$ is dynamic convolutional parameter in $\text{Conv}_{\mathbf{W}, \mathbf{W}'}$, $\sigma_{s,t} = \sum_{p=1}^n \sum_{q=1}^n \mathbf{W}'_{p,q} \mathbf{I}'_{s-p, t-q}$ denotes the standard convolution. We omit the channel dimension for simplicity since it does not affect our local adaptiveness. For

efficiency, we always set $m = 3$, and assign the value of n according to the LAM position in ECMA block. Accordingly, n is called the kernel size of $\text{Conv}_{\mathbf{W}, \mathbf{W}'}$.

Now we discuss the local adaptiveness induced from Eqn. (2). Specifically, Eqn. (2) shows that for pixel $\mathbf{I}_{s-i, t-j}$, its convolution parameter $\tilde{\mathbf{W}}_{i,j}$ depends on the $n \times n$ -sized neighboring region around the pixel $\mathbf{I}'_{s-i, t-j}$ of input feature \mathbf{I}' . This is due to the factor $\sigma_{s,t}$ in the convolution kernel $\tilde{\mathbf{W}}$ incorporating the neighboring region of \mathbf{I}' . Meanwhile, since \mathbf{I}' is spatial context of \mathbf{I} , each feature pixel $\mathbf{I}'_{s,t}$ is computed by convolving feature values from pixels around $\mathbf{I}_{s,t}$. Consequently, the convolution parameter $\tilde{\mathbf{W}}_{i,j}$ for $\mathbf{I}_{s-i, t-j}$ depends on the neighboring region around this pixel, within a specific convolution radius. So the local adaptive convolution $\text{Conv}_{\mathbf{W}, \mathbf{W}'}$ can adaptively tune the convolution parameter via learning suitable parameters $\tilde{\mathbf{W}}_{i,j}$ for a specific input region. This approach increases the flexibility of feature processing and enhances the model's responsiveness to local variations in the image.

Efficient Implementation of LAM. The overall LAM, as depicted in Figure 3b, is formulated as follows:

$$\begin{aligned} \mathbf{Q} &= \text{Cat}(\text{Conv}_{\mathbf{W}, \mathbf{W}'}(\mathbf{I}, \mathbf{I}'), \text{Conv}_{\mathbf{W}}(\mathbf{I}), \text{Conv}_{\mathbf{W}'}(\mathbf{I}')), \\ \mathbf{O} &= \text{Norm}(\mathbf{Q}), \quad \mathbf{O}' = \text{Conv}_{1 \times 1}(\mathbf{Q}), \end{aligned} \quad (3)$$

where Cat denotes concatenation along the channel dimension, and Norm refers to group normalization [52]. In Eqn. (3), we further use the two kernels \mathbf{W} and \mathbf{W}' to respectively compute two feature maps $\text{Conv}_{\mathbf{W}}(\mathbf{I})$ and $\text{Conv}_{\mathbf{W}'}(\mathbf{I})$ for two reasons. First, the convolutions $\text{Conv}_{\mathbf{W}'}$ and $\text{Conv}_{\mathbf{W}}$, using different kernel sizes, capture spatial information at different scales, complementing the local adaptive convolution. Second, Eqn. (1) can be rewritten in its equivalent formulation: $\text{Conv}_{\mathbf{W}, \mathbf{W}'}(\mathbf{I}, \mathbf{I}') = \text{Conv}_{\mathbf{W}}(\mathbf{I}) \odot \text{Conv}_{\mathbf{W}'}(\mathbf{I}')$ with element-wise product \odot (See Appendix A for details). So computing $\text{Conv}_{\mathbf{W}}(\mathbf{I})$ and $\text{Conv}_{\mathbf{W}'}(\mathbf{I})$ does not bring extra computational cost.

Next, following Eqn. (3), we concatenate $\text{Conv}_{\mathbf{W}}(\mathbf{I})$ and $\text{Conv}_{\mathbf{W}'}(\mathbf{I})$ with feature $\text{Conv}_{\mathbf{W}, \mathbf{W}'}(\mathbf{I}, \mathbf{I}')$ given by local adaptive convolution. Finally, we normalize the concatenated feature \mathbf{Q} to obtain \mathbf{O} , which is then processed by a 1×1 convolution $\text{Conv}_{1 \times 1}$ to reduce the channel dimension. LAM has two outputs, \mathbf{O} and \mathbf{O}' . \mathbf{O} is concatenated with the outputs of other LAMs at the end of the ECMA block (see Figure 3a). \mathbf{O}' is used as spatial context for the subsequent LAM, which will be elaborated in Sec. 3.2.

Compared with complex and expensive mechanisms like self-attention, LAM achieves local adaptiveness via cheap local adaptive convolution. Concretely, we compare the computation complexity of our LAM with self-attention operation which also achieve local adaptive processing. For an input with spatial size $s \times s$ and channel number c , self-attention with h heads costs $(2s^2c + 4c^2 + 2s^2h)s^2$, while our LAM with kernel size k only costs $((k^2 + 10)c + 3c^2)s^2$ (see Appendix B for details). Our LAM requires much less computation cost, since the spatial size s is often much larger than the kernel size k .

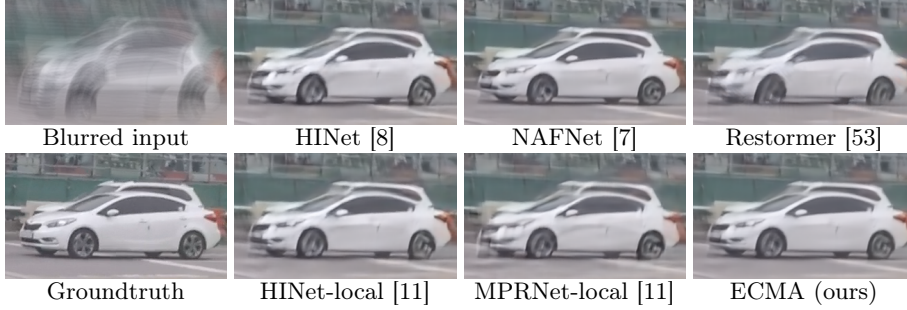


Fig. 4: Qualitative comparison of image deblurring results on GoPro dataset [37].

Cascaded Multiscale Learning In various image restoration tasks, multi-scale feature processing is essential for effectively learning contextual information and addressing diverse scales of degradation. Motion blurring, covering large image areas, benefits from large-scale feature processing [48], whereas scenarios like non-uniform noise or super-resolution of small details require effective local feature processing [7]. To tackle this, we present the Cascaded Multiscale Learning (CML) mechanism which cascades three LAMs with different kernel sizes for efficient multiscale feature learning.

As shown in Fig. 3a, CML first expands the channel number c of input \mathbf{X} to $9c/4$ via an 1×1 convolution, and separates the features into six parts along the channel dimension:

$$\mathbf{I}_1, \mathbf{I}'_1, \mathbf{I}_2, \mathbf{I}'_2, \mathbf{I}_3, \mathbf{I}'_3 = \text{Split}(\text{Conv}_{1 \times 1}(\mathbf{X})). \quad (4)$$

Next, CML adopts three LAMs in a cascading manner

$$\begin{aligned} \mathbf{O}_1, \mathbf{O}'_1 &= \text{LAM}_1(\mathbf{I}_1, \mathbf{I}'_1), \\ \mathbf{O}_2, \mathbf{O}'_2 &= \text{LAM}_2(\mathbf{I}_2, \text{Cat}(\mathbf{O}'_1, \mathbf{I}'_2)), \\ \mathbf{O}_3, \mathbf{O}'_3 &= \text{LAM}_3(\mathbf{I}_3, \text{Cat}(\mathbf{O}'_2, \mathbf{I}'_3)), \end{aligned}$$

where LAM_i ($1 \leq i \leq 3$) denotes LAM module defined in Eqn. (3). For $\text{LAM}_i(\mathbf{I}, \mathbf{C})$, \mathbf{I} is the input, and \mathbf{C} is its associated spatial context (See Sec. 3.2 for details). Finally, CML reduces the channel number of the concatenated output to c , and adds it to the input \mathbf{X} as a residual:

$$\mathbf{Y} = \mathbf{X} + \gamma \cdot \text{Conv}_{1 \times 1}(\text{Cat}(\mathbf{O}_1, \mathbf{O}_2, \mathbf{O}_3)),$$

where $\gamma \in \mathbb{R}^c$ is a learnable vector [47].

Now we discuss how to set the kernel sizes in three LAMs and the splitting in Eqn. (4). Here, we first use a large kernel size followed by smaller ones in the three LAMs to enable comprehensive spatial context while maintaining efficiency. For instance, in LAM_2 , the context \mathbf{O}'_1 in $\text{Cat}(\mathbf{O}'_1, \mathbf{I}'_2)$ is computed via a LAM of large convolution size, providing broad spatial context to the input \mathbf{I}_2 , while the raw feature \mathbf{I}'_2 provides more local spatial context for complementation. In LAM_3 , the context \mathbf{O}'_2 in $\text{Cat}(\mathbf{O}'_2, \mathbf{I}'_3)$, which already contains large spatial context, and thus help guide the smaller LAM to better process local features.

Table 1: ECMA Network configurations. The #Blocks denotes the number of ECMA blocks in each stage in the encoder. The number of ECMA blocks in the decoder stages symmetrically mirrors that of the corresponding encoder stages. Computation complexities are measured on input size 256×256 . *The ECMA-SR model is configured for super-resolution, the GFLOPs is computed for $4 \times$ upscaling to size of 1280×720 .

Variant	#Blocks	Width	M.Par	GFLOPs
ECMA-T	1,1,2,9	32	10.5	15.9
ECMA-S	1,1,1,7	64	31.7	48.4
ECMA-B	2,4,4,3	64	19.5	63.0
ECMA-SR*	10,1	64	1.4	50.5

For $\text{LAM}_i(\mathbf{I}_i, \mathbf{C}_i)$ ($1 \leq i \leq 3$), the kernel sizes of convolution applied on \mathbf{C}_i decreases as aforementioned. Accordingly, we set channel number $c/4, c/2, 3c/4$ for \mathbf{I}'_i ($1 \leq i \leq 3$), and $c/4$ for all \mathbf{I}_i ($1 \leq i \leq 3$) to improve overall efficiency.

CML provides effective multiscale feature learning for image restoration tasks. The first LAM, LAM_1 , aggregates global context through a large receptive field, while the subsequent LAMs, LAM_2 and LAM_3 , refine features at finer scales with the help of the global context provided by LAM_1 . This novel design allows the model to adeptly handle diverse scales of degradation, and resolves variations in artifacts, guided by extensive contextual insights. Consequently, the CML design facilitates a more comprehensive representation of the multiscale characteristics, thereby improving the network’s performance in restoring images affected by complex and diverse degradations. The effectiveness of the CML design is verified in Section 4.5.

To enhance computational efficiency, we make use of Fast Fourier Transform (FFT) [4] in the LAM_1 of large kernels. This approach transforms image features into frequency components, simplifying convolutions to multiplications. It cuts complexity from $O(n^2)$ to $O(n \log n)$, where n is the spatial size. This method reduces the computational cost, especially for large kernel size convolutions in the large LAMs, enhancing our CML’s effectiveness and efficiency to learn large-scale context for image restoration.

3.3 Fusion Block

The fusion block fuses the previous features from the encoder with the current upsampled features. It first upsamples the decoder feature $\bar{\mathbf{D}}_{i-1}$ via transposed convolution with a kernel and stride size of 2×2 , concatenates it with the encoder input \mathbf{E}_i , and finally fuses them using a 1×1 convolution. Formally, to compute the feature \mathbf{D}_i for the next stage in the decoder, the fusion block is formulated as: $\mathbf{D}_i = \text{Conv}_{1 \times 1}(\text{Cat}(\text{TConv}(\bar{\mathbf{D}}_{i-1}), \mathbf{E}_i))$, where TConv denotes transposed convolution.

3.4 ECMA Network

The ECMA networks employ a U-Net architecture, with our efficient ECMA and fusion blocks, tailored for image restoration tasks. We build ECMA networks in different sizes, as shown in Table 1. Each ECMA block contains three LAMs

Table 2: Image Deblurring results on GoPro dataset.

Method	GFLOPs	PSNR	SSIM
DeblurGAN-v2 [20]	—	29.55	0.934
SRN [45]	—	30.25	0.934
DMPHN [57]	—	31.20	0.945
SimpleNet [22]	—	31.52	0.950
SAPHN [43]	—	32.02	0.953
MIMO [10]	1235	32.45	0.957
IPT [6]	—	32.58	-
MPRNet [55]	778	32.66	0.959
Restormer [53]	140	32.92	0.961
Uformer-B [51]	89.5	33.06	0.967
Stripformer [48]	155	33.08	0.962
ECMA-T (Ours)	15.9 (9.7×↓)	33.19	0.962
DeepRFT+ [34]	187	33.23	0.963
MSSNet-L [18]	—	33.39	0.964
NAFNet [7]	65	33.69	0.967
GRL-B [24]	281	33.93	0.968
ECMA-B (Ours)	63.0 (4.5×↓)	34.14	0.969

with adaptive convolution kernel sizes: k_i , 5, and 3. The k_i size begins at 32 and reduces by 8 at each encoder stage. Conversely, the kernel sizes k_i increase along the stages in the decoder, mirroring the encoder.

The innovative integration of LAM and CML in ECMA facilitates the effective reconstruction of high-quality images from degraded inputs. ECMA networks achieve comparable or superior restoration performance to current leading methods, while significantly reducing computational costs by $1.2\times$ to $9.7\times$ across various image restoration tasks, as shown in Figure 1.

4 Experiments

Training Configuration. We train our models with a batch size of 64 for 400K iterations, using the AdamW optimizer [30] with an initial learning rate of 1×10^{-3} and a cosine annealing strategy to a final learning rate of 1×10^{-7} . The spatial size during training is 256×256 . We perform data augmentations, including random cropping, flipping, and rotation, following [7, 48].

Training Loss. Our training loss, similar to previous works [10], combines L1-Loss in pixel space for pixel-wise reconstruction and L1-Loss in frequency space for frequency alignment (Refer to Appendix A for details). We empirically set the weights for pixel loss and frequency loss to 1.0 and 0.05 respectively.

Evaluation metrics. We employ two commonly used metrics for evaluation: Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM). PSNR serves as an objective error measure, while SSIM provides a perceptually-aligned assessment. Higher values in these metrics indicate better restoration performance.

4.1 Deblurring Results

Deblurring Results on GoPro We follow the protocols in [37] and evaluate ECMA on the widely used GoPro dataset. Table 2 shows that ECMA achieves

Table 3: Image Deblurring results on the HIDE dataset.

Method	GFLOPs	PSNR	SSIM
DeblurGAN-v2 [20]	—	26.61	0.875
SRN [45]	—	28.36	0.904
SAPHN [44]	—	29.98	0.930
MIMO-Unet+ [10]	1235	30.00	0.930
Uformer-B [51]	89.5	30.90	0.953
ECMA-T (Ours)	15.9 ($5.6\times\downarrow$)	30.89	0.937
MPRNet [55]	—	30.96	0.940
Stripformer [48]	155	31.03	0.940
MPRNet-local [11]	778	31.19	0.942
Restormer [53]	140	31.22	0.942
NAFNet [7]	65	31.31	0.943
Restormer-local [11]	140	31.49	0.945
ECMA-B (Ours)	63.0 ($2.2\times\downarrow$)	31.59	0.946

the highest PSNR and SSIM values, outperforming existing methods. Compared to state-of-the-art methods like GRL-B [24], ECMA achieves a PSNR improvement of 0.21dB, while using significantly less computation, i.e. the FLOPs of GRL-B is $4.5\times$ that of our ECMA-B. Furthermore, compared to CNN representative methods like NAFNet [7], ECMA shows superior performance while requiring only about 30% of NAFNet’s model size. The results reveal that ECMA not only attains higher restoration performance but also excels in terms of efficiency. Figure 4 provides visual comparisons between ECMA and other methods on the GoPro dataset. By observing, Observation reveals that CNN-based methods [7, 8] suffer from noticeable blur issues, as they may struggle to effectively explore non-local information for image restoration. Transformer-based methods, such as [48, 53], also fail to recover certain local details. In contrast, ECMA introduces an efficient local adaptive and multiscale learning approach, better processing features of blur effects at different scales for image restoration. These results demonstrate ECMA’s effectiveness and superiority in handling varying blur scales and its computational efficiency.

Deblurring Results on HIDE We further test ECMA on the HIDE dataset, which is known for its emphasis on human subjects [42]. For fair comparison, we follow [7, 10, 55], and apply our model trained on the GoPro dataset to the HIDE dataset for evaluation. Table 3 shows that ECMA achieves a PSNR of 31.59 dB and an SSIM of 0.946, offering higher deblurring quality than other evaluated methods. Notably, ECMA significantly improves efficiency compared to the state-of-the-art, Restormer-local [53], by only costing $2.2\times$ fewer FLOPs to achieve better performance.

Deblurring Results on RealBlur We evaluated our ECMA on the RealBlur dataset [39], which contains images captured under various conditions, including low-light and motion scenarios. The RealBlur dataset includes two test sets: RealBlur-R, derived from raw images, and RealBlur-J, sourced from JPEG images. For fair comparison, we followed the evaluation settings of [7, 48]. Table 4 shows that ECMA achieves a PSNR of 32.72dB and 39.98dB on RealBlur-J and

Table 4: Image Deblurring results on RealBlur datasets.

Method	GFLOPs	RealBlur-J		RealBlur-R	
		PSNR	SSIM	PSNR	SSIM
DeblurGANv2 [20]	—	29.69	0.870	36.44	0.935
Restormer [53]	140	28.96	0.879	36.19	0.957
SRN [45]	—	31.38	0.909	38.65	0.965
MPRNet [55]	778	31.76	0.922	39.31	0.972
MIMO-UNet+ [10]	1235	32.05	0.921	-	-
BANet [49]	—	32.00	0.923	39.55	0.971
MSSNet [18]	—	32.10	0.928	39.76	0.972
DeepRFT+ [34]	187	32.19	0.931	39.84	0.972
Stripformer	155	32.48	0.929	39.84	0.974
ECMA-B (ours)	63.0 (2.5×↓)	32.72	0.931	39.98	0.974

RealBlur-R, respectively. For efficiency, ECMA requires only 63 GFLOPs, $2.5\times$ reduction from Stripformer’s 155 GFLOPs.

4.2 Image Super-Resolution Results

Our experiments extend to the task of image super-resolution, where we assess the ECMA network’s performance on established public benchmark datasets. Super-resolution, a critical aspect of image restoration, requires precise up-scaling of image details from low to high resolution. We perform training on DIV2K [26] and evaluation following recent works [53, 62]. In Table 5, we present results for image super-resolution with upscaling factor of $4\times$. Computation costs are measured based on a target image size of 1280×720 . The ECMA-SR network, designed for efficiency, consumes around $1.2\times$ fewer FLOPs than the SRFormer [62], while delivering comparable or superior PSNR and SSIM metrics. This balance of reduced computational demand and state-of-the-art performance underscores the ECMA network’s capability in super-resolution, reinforcing its applicability in image restoration.

4.3 JPEG Artifacts Results

To assess the robustness of our ECMA network in handling real-world scenarios, we conducted deblurring tests on the REDS dataset, which includes realistic video sequences with dynamic scenes and JPEG compression artifacts. This dataset is particularly challenging due to its inclusion of various blur types and compression-induced distortions, providing a rigorous benchmark for evaluating our model’s adaptability. Following established protocols from prior studies [7, 8, 50], we trained our network and evaluated it on a subset of 300 images (REDS-val-300) from the REDS validation set. Table 6 shows the performance of our ECMA networks compared to other state-of-the-art works. ECMA-T achieves competitive PSNR of 28.82dB, similar to HINet [8], while being 10 times smaller in computation cost. Moreover, our ECMA-S variant achieves state-of-the-art performance with 29.12dB in PSNR and 0.868 in SSIM. Notably, our ECMA-S model requires $1.3\times$ fewer FLOPs than the NAFNet model, further demonstrating the efficiency of our ECMA design.

Table 5: Image Super-resolution results for $4\times$ upscaling. All models are trained on DIV2k [26] and evaluated on [3, 16, 35, 36, 56].

Method	GFLOPs	Set5 [3]		Set14 [56]		B100 [35]		U100 [16]		M109 [36]	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
EDSR-baseline [26]	114	32.09	0.894	28.58	0.781	27.57	0.736	26.04	0.785	30.35	0.907
CARN [2]	90.9	32.13	0.894	28.60	0.781	27.58	0.735	26.07	0.784	30.47	0.908
IMDN [17]	40.9	32.21	0.895	28.58	0.781	27.56	0.735	26.04	0.784	30.45	0.907
LAPAR-A [23]	94	32.15	0.894	28.61	0.782	27.61	0.737	26.14	0.787	30.42	0.907
RFDN [27]	—	32.28	0.896	28.61	0.782	27.58	0.736	26.20	0.788	30.61	0.910
LatticeNet [32]	43.6	32.30	0.896	28.68	0.783	27.62	0.737	26.25	0.787	-	-
ESRT [31]	—	32.19	0.895	28.69	0.783	27.69	0.738	26.39	0.796	30.75	0.910
SwinIR-light [25]	63.6	32.44	0.898	28.77	0.786	27.69	0.741	26.47	0.798	30.92	0.915
ELAN [59]	54.1	32.43	0.897	28.78	0.786	27.69	0.741	26.54	0.798	30.92	0.915
SRFormer [62]	62.8	32.51	0.899	28.82	0.787	27.73	0.742	26.67	0.803	31.17	0.916
ECMA-SR	50.5(1.2 \times ↓)	32.50	0.899	28.89	0.788	27.81	0.744	26.82	0.806	31.20	0.917

4.4 Denoising Results

In addition to image deblurring, we perform experiments of RGB Image Denoising on the Smartphone Image Denoising (SIDD) dataset [1]. The SIDD dataset is unique in its focus on noise patterns found in smartphone photography, providing a benchmark for practical denoising applications. As shown in Table 7, the ECMA-T model attains a PSNR of 39.86dB with only 15.9 GFLOPs, which is $5.6\times$ reduction on UFormer’s [51] computational cost. The ECMA-S model demonstrates competitive performance with a PSNR of 40.28 dB and an SSIM of 0.961, on par with the advanced NAFNet [7], while achieving a $1.3\times$ reduction in computational complexity. These results highlight the ECMA network’s ability to provide state-of-the-art denoising efficiently, further confirming its effectiveness and generalizability for diverse image restoration applications.

4.5 Ablation Study

For ablation studies, we train all variants of ECMA-T models for 200K iterations with batch size of 64 on the GoPro dataset.

Investigation on LAM We achieve efficient local adaptive feature processing in our method with LAM, via local adaptive convolution. This design choice is central to our method’s ability to adapt to varying local features within an image, thereby enhancing its restoration capabilities. To verify the effectiveness of this local adaptive mechanism, we conduct an ablation study. In this study, we replace our proposed local adaptive feature processing with alternative methods for combining features from two input branches. The results of this ablation study are presented in Table 8. They reveal a clear advantage of our local adaptive design. When local adaptiveness is removed or replaced with another operation (element-wise addition), performance suffers noticeably. Specifically, the PSNR drops by as much as 0.62dB and 0.68dB respectively. These findings strongly validate our design choice, demonstrating that the local adaptive feature processing is a fundamental component that contributes to ECMA’s effectiveness. It allows our model to respond to the unique characteristics of different regions within an image, leading to more accurate restoration results.

Table 6: Deblurring with JPEG artifacts removal on REDS dataset.

Method	GFLOPs	PSNR	SSIM
MPRNet [55]	777	28.79	0.811
HINet [8]	171	28.83	0.862
ECMA-T (Ours)	15.9(10.8 \times ↓)	28.82	0.861
MAXIM [50]	170	28.93	0.865
NAFNet [7]	65	29.09	0.867
ECMA-S (Ours)	48.4(1.3 \times ↓)	29.12	0.868

Table 7: Denoising on the SIDD.

Method	GFLOPs	PSNR	SSIM
MPRNet [55]	588	39.71	0.958
MIRNet [54]	786	39.72	0.959
NBNet [9]	88.8	39.75	0.959
UFormer [51]	89.5	39.85	0.960
ECMA-T (Ours)	15.9(5.6 \times ↓)	39.86	0.960
MAXIM [50]	170	39.96	0.960
HINet [8]	171	39.99	0.958
NAFNet [7]	65	40.30	0.962
ECMA-S (Ours)	48.4(1.3 \times ↓)	40.28	0.961

Table 8: Ablation for LAM.

Variant	PSNR	SSIM
Local Adaptive Conv (Proposed)	32.98	0.961
Ele-wise Addition	32.36	0.955
Direct concatenation	32.30	0.954

Table 9: Ablation for CML.

Variant	PSNR	SSIM
cascaded large to small LAM (Proposed)	32.98	0.961
cascaded small to large LAM	32.64	0.957
multiscale without cascading	32.56	0.956
without large LAM	32.28	0.955
without medium LAM	32.69	0.958
without small LAM	32.61	0.957
cascade 3 large LAMs	32.29	0.955
cascade 3 medium LAMs	32.51	0.956
cascade 3 small LAMs	32.26	0.955
large conv in spatial domain	32.47	0.955

Investigation on CML We further conduct an ablation study on the efficient cascaded multiscale learning, another key aspect of the ECMA block’s design. This cascaded multiscale feature processing allows our method to effectively utilize contextual information and handle varying distortion scales, and is instrumental in achieving our method’s efficient performance. To assess the contribution of the cascade multiscale design, we conduct a comprehensive ablation study. To ensure a fair comparison, we modify block and channel numbers to align the model size and complexity with our original proposed design. The results of this ablation study are shown in Table 9. Notably, changing the order of the cascaded design or removing the cascading mechanism completely would result in a decrease in PSNR by 0.34dB and 0.42dB, respectively. We also experimented with several variants by selectively removing one of the branches, each would incur a loss of performance. We performed further ablations with cascading 3 identical LAMs. The results in table 9 shows performances obtained from using identical LAMs of different sizes are inferior than our cascade design with large to small LAMs. Moreover, performing the large convolution the standard way in the spatial domain, while keeping computation cost at the same level, leads to a substantial reduction in PSNR by more than 0.51dB. These ablation results demonstrate the effectiveness and efficiency of our cascaded multiscale learning in ECMA for image restoration.

4.6 Inference Time

Results have shown that ECMA models require greatly reduced computation FLOPs compared to other state-of-the-art methods. Here, we measure and compare the actual inference time for various image deblurring models on a single GPU. Table 10 shows the results on single Nvidia A100 GPU, we take the average inference time of 10 runs with input size of 1280×720 for each model.

Table 10: Single image inference time comparison.

Model	FLOPS	Time	Model	FLOPS	Time
Uformer-B	89.5G	42 ms	ECMA-T (ours)	15.9G	26 ms
DeepRFT+	187G	54 ms	ECMA-S (ours)	48.4G	27 ms
Restormer	140G	59 ms	ECMA-B (ours)	63.0G	39 ms

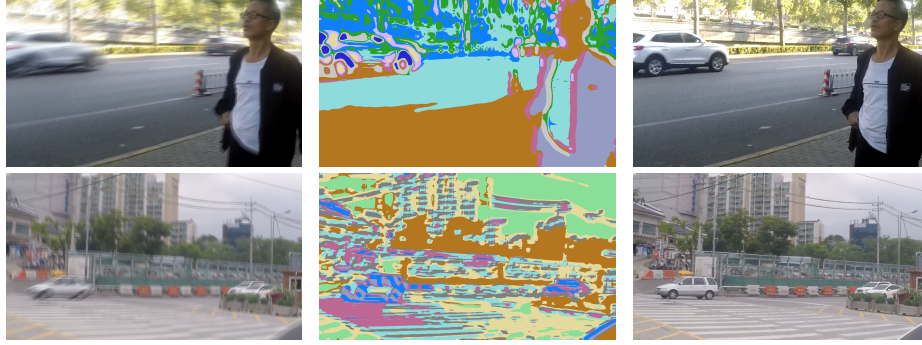


Fig. 5: Visualization of adaptive kernels (middle) for restoring input with motion blur (left) to sharp image (right), regions with the same color have similar local adaptive kernels \bar{W} generated in our LAM .

4.7 Visualization

To verify the effectiveness of the proposed Local Adaptive Module, we visualize the kernels generated from a trained ECMA model. Specifically, we perform clustering on the generated adaptive kernels parameters \bar{W} in Eqn. (2) from trained ECMA-B model. Fig. 5 shows the visualization of clustering, where each color represents a cluster of similar kernels parameters. We can see varying adaptive kernels across an image, and similar kernels within a region of similar blurring effects. This shows that our LAM can effectively generate dynamic kernels that adapt to local features and degradations.

5 Conclusion

In this paper, we present Efficient Cascaded Multiscale Adaptive (ECMA), a new performant and efficient approach for image restoration. ECMA combines local adaptive processing with cascaded multiscale learning to effectively handle diverse image degradations while maintaining computational efficiency. Our extensive experiments on standard benchmarks shows that ECMA achieves top-tier performance with lower computational costs compared to existing methods. ECMA’s ability to accurately address spatially varying artifacts at a reduced computational expense makes it a valuable tool for various image restoration tasks. Future research could extend ECMA’s application to other areas, potentially broadening its impact in the field. This work offers a efficient perspective on image restoration and promote further innovations in adaptive feature processing in computer vision.

Acknowledgements

Pan Zhou was supported by the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 grant. Yichen Zhou was supported by the Economic Development Board Industrial Postgraduate Programme. Pan Zhou is the corresponding author.

References

1. Abdelhamed, A., Lin, S., Brown, M.S.: A high-quality denoising dataset for smart-phone cameras. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1692–1700 (2018)
2. Ahn, N., Kang, B., Sohn, K.A.: Fast, accurate, and lightweight super-resolution with cascading residual network. In: ECCV (2018)
3. Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.L.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding (2012)
4. Brigham, E.O.: The fast Fourier transform and its applications. Prentice-Hall, Inc. (1988)
5. Burger, H.C., Schuler, C.J., Harmeling, S.: Image denoising: Can plain neural networks compete with bm3d? In: 2012 IEEE conference on computer vision and pattern recognition. pp. 2392–2399. IEEE (2012)
6. 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: CVPR (2021)
7. Chen, L., Chu, X., Zhang, X., Sun, J.: Simple baselines for image restoration. In: ECCV (2022)
8. 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)
9. Cheng, S., Wang, Y., Huang, H., Liu, D., Fan, H., Liu, S.: Nbnnet: 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)
10. 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: ICCV (2021)
11. Chu, X., Chen, L., , Chen, C., Lu, X.: Improving image restoration by revisiting global information aggregation. In: ECCV (2022)
12. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence **38**(2), 295–307 (2015)
13. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
14. Fedorov, V., Ballester, C.: Affine non-local means image denoising. IEEE Transactions on Image Processing **26**(5), 2137–2148 (2017)
15. Hu, X., Ren, W., Yu, K., Zhang, K., Cao, X., Liu, W., Menze, B.: Pyramid architecture search for real-time image deblurring. In: ICCV (2021)
16. Huang, J.B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: CVPR (2015)

17. Hui, Z., Gao, X., Yang, Y., Wang, X.: Lightweight image super-resolution with information multi-distillation network. In: ACM MM (2019)
18. Kim, K., Lee, S., Cho, S.: Mssnet: Multi-scale-stage network for single image deblurring. In: European Conference on Computer Vision. pp. 524–539. Springer (2022)
19. Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., Matas, J.: Deblurgan: Blind motion deblurring using conditional adversarial networks. In: CVPR (2018)
20. Kupyn, O., Martyniuk, T., Wu, J., Wang, Z.: Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In: ICCV (2019)
21. Levin, A., Fergus, R., Durand, F., Freeman, W.T.: Deconvolution using natural image priors. Massachusetts Institute of Technology, Computer Science and Artificial Intelligence Laboratory **3** (2007)
22. Li, J., Tan, W., Yan, B.: Perceptual variousness motion deblurring with light global context refinement. In: ICCV (2021)
23. Li, W., Zhou, K., Qi, L., Jiang, N., Lu, J., Jia, J.: Lapar: Linearly-assembled pixel-adaptive regression network for single image super-resolution and beyond. In: NeurIPS (2020)
24. Li, Y., Fan, Y., Xiang, X., Demandolx, D., Ranjan, R., Timofte, R., Van Gool, L.: Efficient and explicit modelling of image hierarchies for image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18278–18289 (2023)
25. Liang, J., Cao, J., Sun, G., Zhang, K., Gool, L.V., Timofte, R.: Swinir: Image restoration using swin transformer. In: ICCV Workshops (2021)
26. Lim, B., Son, S., Kim, H., Nah, S., Mu Lee, K.: Enhanced deep residual networks for single image super-resolution. In: CVPR Workshops (2017)
27. Liu, J., Tang, J., Wu, G.: Residual feature distillation network for lightweight image super-resolution. In: Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16. pp. 41–55. Springer (2020)
28. Liu, X., Tanaka, M., Okutomi, M.: Single-image noise level estimation for blind denoising. IEEE transactions on image processing **22**(12), 5226–5237 (2013)
29. 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: ICCV (2021)
30. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization (2019)
31. Lu, Z., Liu, H., Li, J., Zhang, L.: Efficient transformer for single image super-resolution. arXiv:2108.11084 (2021)
32. Luo, X., Xie, Y., Zhang, Y., Qu, Y., Li, C., Fu, Y.: Latticenet: Towards lightweight image super-resolution with lattice block. In: ECCV (2020)
33. Mao, X., Shen, C., Yang, Y.B.: Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. Advances in neural information processing systems **29** (2016)
34. Mao, X., Liu, Y., Shen, W., Li, Q., Wang, Y.: Deep residual fourier transformation for single image deblurring. arXiv preprint arXiv:2111.11745 (2021)
35. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV (2001)
36. Matsui, Y., Ito, K., Aramaki, Y., Fujimoto, A., Ogawa, T., Yamasaki, T., Aizawa, K.: Sketch-based manga retrieval using manga109 dataset. Multimedia Tools and Applications (2017)
37. Nah, S., Kim, T.H., Lee, K.M.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: CVPR (2017)

38. Park, D., Kang, D.U., Kim, J., Chun, S.Y.: Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training. In: ECCV (2020)
39. Rim, J., Lee, H., Won, J., Cho, S.: Real-world blur dataset for learning and benchmarking deblurring algorithms. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16. pp. 184–201. Springer (2020)
40. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015)
41. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. *Acm transactions on graphics (tog)* **27**(3), 1–10 (2008)
42. Shen, Z., Wang, W., Lu, X., Shen, J., Ling, H., Xu, T., Shao, L.: Human-aware motion deblurring. In: ICCV (2019)
43. Suin, M., Purohit, K., Rajagopalan, A.N.: Spatially-attentive patch-hierarchical network for adaptive motion deblurring. In: CVPR (2020)
44. Suin, M., Purohit, K., Rajagopalan, A.N.: Spatially-attentive patch-hierarchical network for adaptive motion deblurring. In: CVPR (2020)
45. Tao, X., Gao, H., Shen, X., Wang, J., Jia, J.: Scale-recurrent network for deep image deblurring. In: CVPR (2018)
46. Tian, C., Xu, Y., Zuo, W.: Image denoising using deep cnn with batch renormalization. *Neural Networks* **121**, 461–473 (2020)
47. Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., Jégou, H.: Going deeper with image transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 32–42 (2021)
48. Tsai, F.J., Peng, Y.T., Lin, Y.Y., Tsai, C.C., Lin, C.W.: Stripformer: Strip transformer for fast image deblurring. In: ECCV (2022)
49. Tsai, F.J., Peng, Y.T., Tsai, C.C., Lin, Y.Y., Lin, C.W.: Banet: a blur-aware attention network for dynamic scene deblurring. *IEEE Transactions on Image Processing* **31**, 6789–6799 (2022)
50. Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A., Li, Y.: Maxim: Multi-axis mlp for image processing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5769–5780 (2022)
51. Wang, Z., Cun, X., Bao, J., Zhou, W., Liu, J., Li, H.: Uformer: A general u-shaped transformer for image restoration. In: CVPR (2022)
52. Wu, Y., He, K.: Group normalization. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018)
53. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H.: Restormer: Efficient transformer for high-resolution image restoration. In: CVPR (2022)
54. 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: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16. pp. 492–511. Springer (2020)
55. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Multi-stage progressive image restoration. In: CVPR (2021)
56. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations. In: Curves and Surfaces (2010)
57. Zhang, H., Dai, Y., Li, H., Koniusz, P.: Deep stacked hierarchical multi-patch network for image deblurring. In: CVPR (2019)
58. Zhang, K., Luo, W., Zhong, Y., Ma, L., Stenger, B., Liu, W., Li, H.: Deblurring by realistic blurring. In: CVPR (2020)

- 59. Zhang, X., Zeng, H., Guo, S., Zhang, L.: Efficient long-range attention network for image super-resolution. *arXiv:2203.06697* (2022)
- 60. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: *Proceedings of the European conference on computer vision (ECCV)*. pp. 286–301 (2018)
- 61. Zhou, D., Kang, B., Jin, X., Yang, L., Lian, X., Jiang, Z., Hou, Q., Feng, J.: Deepvit: Towards deeper vision transformer. *arXiv:2103.11886* (2021)
- 62. Zhou, Y., Li, Z., Guo, C.L., Bai, S., Cheng, M.M., Hou, Q.: Srformer: Permuted self-attention for single image super-resolution. *arXiv preprint arXiv:2303.09735* (2023)