On Efficient Transformer-Based Image Pre-training for Low-Level Vision

Wenbo Li[1*], **Xin Lu**[2*], **Shengju Qian**[1] and **Jiangbo Lu**[3]
[1]The Chinese University of Hong Kong
[2]Deeproute.ai
[3]SmartMore Corporation
{wenboli,sjqian}@cse.cuhk.edu.hk, luxin941027@gmail.com, jiangbo.lu@gmail.com

Abstract

Pre-training has marked numerous state of the arts in high-level computer vision, while few attempts have ever been made to investigate how pre-training acts in image processing systems. In this paper, we tailor transformer-based pre-training regimes that boost various low-level tasks. To comprehensively diagnose the influence of pre-training, we design a whole set of principled evaluation tools that uncover its effects on internal representations. The observations demonstrate that pre-training plays strikingly different roles in low-level tasks. For example, pre-training introduces more local information to intermediate layers in super-resolution (SR), yielding significant performance gains, while pre-training hardly affects internal feature representations in denoising, resulting in limited gains. Further, we explore different methods of pre-training, revealing that multi-related-task pre-training is more effective and data-efficient than other alternatives. Finally, we extend our study to varying data scales and model sizes, as well as comparisons between transformers and CNNs. Based on the study, we successfully develop state-of-the-art models for multiple low-level tasks.

1 Introduction

Image pre-training has received great attention in computer vision, especially prevalent in object detection and segmentation [Girshick *et al.*, 2014; Girshick, 2015; Chen *et al.*, 2017]. When task-specific data is limited, pre-training helps models see large-scale data, thus vastly enhancing their capabilities. In the field of high-level vision, previous work [Kornblith *et al.*, 2019; Sun *et al.*, 2017; Mahajan *et al.*, 2018; Kolesnikov *et al.*, 2020] has shown that ConvNets pre-trained on ImageNet [Deng *et al.*, 2009] classification yield significant improvements on a wide spectrum of downstream tasks. As for image processing tasks, e.g., super-resolution (SR) and deraining, the widely used datasets typically contain only a few thousand images, pointing out the potential of pre-training. However, its crucial role in low-level vision is commonly omitted. To the best of our knowledge, the sole pioneer exploring this point is IPT [Chen *et al.*, 2021]. Hence, there still lacks principled analysis on understanding how pre-training acts and how to perform effective pre-training.

Previous image processing systems majorly leverage convolutional neural networks (CNNs) [LeCun *et al.*, 1989]. More recently, transformer architectures [Dosovitskiy *et al.*, 2020; Liu *et al.*, 2021; Wang *et al.*, 2021a], initially proposed in NLP [Vaswani *et al.*, 2017], have achieved promising results in vision tasks, demonstrating the potential of using transformers as a primary backbone for vision applications. Moreover, the stronger modeling capability of transformers allows for large-scale and sophisticated pre-training, which has shown great success in both NLP and computer vision [Radford *et al.*, 2018; Radford *et al.*, 2019; Brown *et al.*, 2020; Devlin *et al.*, 2018; He *et al.*, 2021; Liu *et al.*, 2022; Zamir *et al.*, 2022; Chen *et al.*, 2022]. However, it remains infeasible to directly exploit structure designs and data utilization on the *full-attention* transformers for low-level vision. For example, due to the massive amount of parameters (e.g., 116M for IPT [Chen *et al.*, 2021]) and huge computational cost, it is prohibitively hard to explore various pre-training design choices based on IPT and further apply them in practice. Instead of following the full-attention pipeline, we explore the other *window-based* variants [Liang *et al.*, 2021; Wang *et al.*, 2021b], which are more computationally efficient while leading to impressive performance. Along this line, we develop an encoder-decoder-based transformer (EDT) that is powerful yet efficient in data exploitation and computation. We mainly adopt EDT as a representative for efficient computation, since our observations generalize well to other frameworks, as shown in Sec. 3.4.

In this paper, we systematically explore and evaluate how image pre-training performs in window-based transformers. Using centered kernel alignment [Kornblith *et al.*, 2019; Cortes *et al.*, 2012] as a network "diagnosing" measure, we have designed a set of pre-training strategies, and thoroughly tested them with different image processing tasks. As a result, we uncover their respective effects on internal network representations, and draw useful guidelines for applying pre-training to low-level vision. The key findings and contributions of this study can be summarized as follows,

- **Internal representations of transformers**. We find striking differences in low-level tasks, e.g., SR and de-

*Equal contribution

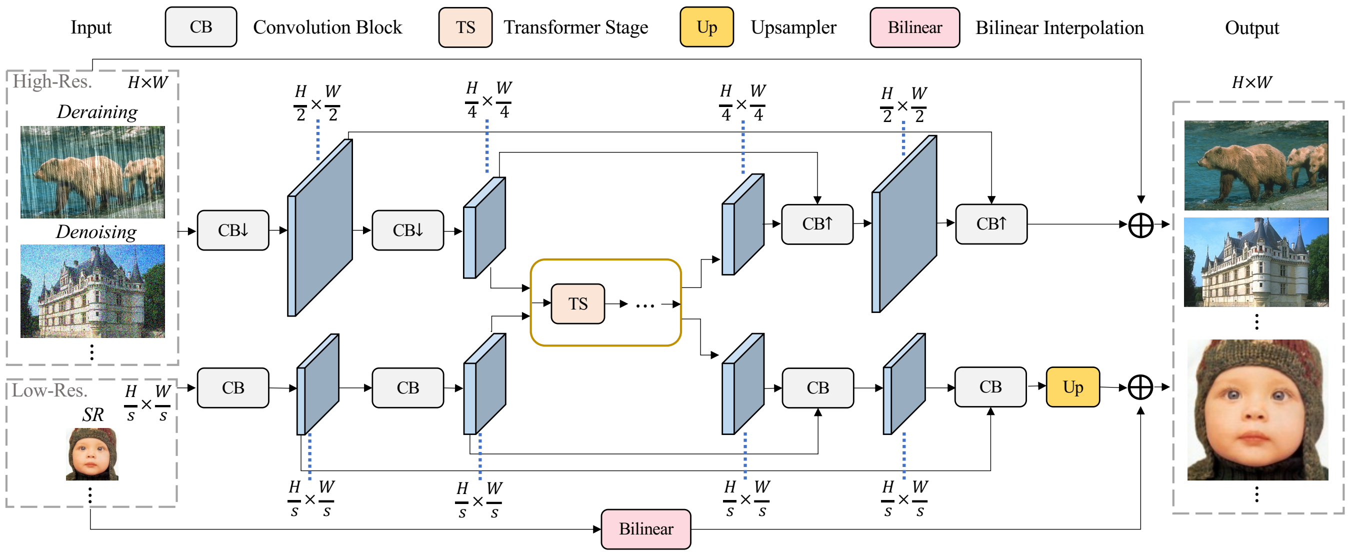

Figure 1: The proposed encoder-decoder-based transformer (EDT). It processes high-resolution (e.g., in denoising) and low-resolution (e.g., in SR, s is the scale) inputs using different paths, modeling long-range interactions at a low resolution, for efficient computation.

raining models show clear stages, containing more local information in early layers while more global information in higher layers. The denoising model presents a relatively uniform structure filled with local information.

- **Effects of pre-training**. We find that pre-training improves the model performance by introducing different degrees of local information, treated as a kind of inductive bias, to the intermediate layers.
- **Pre-training guidelines**. Examining different pre-training strategies, we suggest a favorable *multi-related-task setup* that brings more improvements and could be applied to multiple downstream tasks. Also, we find this performing strategy is more data-efficient than purely increasing the data scale. Besides, a larger model capacity usually gets more out of pre-training.
- **Transformers v.s. CNNs**. We observe that both transformers and CNNs benefit from pre-training, while transformers obtain greater improvements.
- **SOTA models**. Based on the comprehensive study of pre-training, we provide a series of pre-trained models with state-of-the-art performance for multiple tasks, including super-resolution, denoising and deraining.

2 Encoder-Decoder-Based Transformer

Several transformers [Chen *et al.*, 2021; Liang *et al.*, 2021; Wang *et al.*, 2021b] are tailored to low-level tasks, among which window-based architectures [Liang *et al.*, 2021; Wang *et al.*, 2021b] show competitive performance under constrained parameters and computational complexity. Built upon the existing work, we make several modifications and present an efficient encoder-decoder-based transformer (EDT) in Fig. 1. It achieves state-of-the-art results on multiple low-level tasks (see Sec. 4), especially for those with heavy degradation. For example, EDT yields *0.49*dB improvement in $\times 4$ SR on the Urban100 [Huang *et al.*, 2015] benchmark compared to IPT, while our $\times 4$ SR model size (11.6M) is only *10.0%* of IPT (115.6M) and only requires 200K images (*15.6%* of IPT) for pre-training. Also, our denoising model obtains superior performance in level-50 Gaussian denoising, with 38 GFLOPs for 192×192 inputs, far less than SwinIR [Liang *et al.*, 2021] (451 GFLOPs), accounting for only *8.4%*. And the inference speed of EDT (51.9ms) is much faster than SwinIR (271.9ms). It should be pointed out that designing a novel framework is not our main purpose. Noticing similar pre-training effects on transformers in Sec. 3.4, we adopt EDT for fast pre-training in this paper.

2.1 Overall Architecture

As shown in Fig. 1, our EDT is composed of a lightweight convolutional encoder and decoder as well as a transformer-based body, for modeling long-range interactions.

To improve the encoding efficiency, images are first downsampled to $1/4$ size with strided convolutions for tasks with high-resolution inputs (e.g., denoising or deraining), while being processed under the original size for those with low-resolution inputs (e.g., SR). The stack of early convolutions is also proven useful for stabling the optimization [Xiao *et al.*, 2021]. Then, there follow multiple stages of transformer blocks, achieving a large receptive field at a low computational cost. It is noted that we improve the structure of transformer blocks through a series of ablations and provide more details in the supplementary file. During the decoding phase, we upsample the feature back to the input size using transposed convolutions for denoising or deraining while maintaining the size for SR. Besides, skip connections are introduced to enable fast convergence during training. In particular, there is an additional convolutional upsampler before the output for super-resolution.

2.2 Architecture Variants

We develop four variants of EDT with different model sizes, rendering our framework easily applied in various scenarios.

Models	EDT-T	EDT-S	EDT-B	EDT-L
#Channels	60	120	180	240
#Stages	4	5	6	12
#Heads	6	6	6	8
#Param. ($\times 10^6$, M)	0.9	4.2	11.5	40.2
FLOPs ($\times 10^9$, G)	2.8	12.4	37.6	136.4

Table 1: Configurations of four variants of EDT. The parameter numbers and FLOPs are counted in denoising at 192×192 size.

As shown in Table 1, apart from the base model (EDT-B), we also provide EDT-T (Tiny), EDT-S (Small) and EDT-L (Large). The main differences lie in the channel number, stage number and head number in the transformer body. We uniformly set the block number in each transformer stage to 6, the expansion ratio of the feed-forward network (FFN) to 2 and the window size to $(6, 24)$.

3 Study of Image Pre-training

3.1 Pre-training on ImageNet

Following [Chen *et al.*, 2021], we adopt the ImageNet [Deng *et al.*, 2009] dataset in the pre-training stage. Unless specified otherwise, we only use *200K* images for fast pre-training. We choose three representative low-level tasks including super-resolution (SR), denoising and deraining. Referring to [Chen *et al.*, 2021; Agustsson and Timofte, 2017; Gu *et al.*, 2017], we simulate the degradation procedure to synthesize low quality images. In terms of SR, we utilize bicubic interpolation to obtain low-resolution images. As for denoising and deraining, Gaussian noises (on RGB space) and rain streaks are directly added to the clean images. In this work, we explore $\times 2/\times 3/\times 4$ settings in SR, 15/25/50 noise levels in denoising and light/heavy rain streaks in deraining.

We explore three pre-training strategies: *on a single task, on unrelated tasks* and *on related tasks*. (1) Single-task pre-training refers to training a single model on a specific task (e.g., $\times 2$ SR). (2) The second is to train a single model on multiple yet unrelated tasks (e.g., $\times 2$ SR, level-15 denoising), while (3) the last contains highly related tasks (e.g., $\times 2$, $\times 3$ SR). Following [Chen *et al.*, 2021], we adopt a multi-encoder, multi-decoder, shared-body architecture for the latter two setups. The fine-tuning is performed on a single task, where the model is initialized with the pre-trained task-specific encoder and decoder as well as the shared transformer body. Training details are provided in the supplementary file.

3.2 Centered Kernel Alignment

We introduce centered kernel alignment (CKA)[Kornblith *et al.*, 2019; Cortes *et al.*, 2012; Raghu *et al.*, 2021] to study representation similarity of network hidden layers, supporting quantitative comparisons within and across networks. In detail, given m data points, we calculate the activations of two layers $\mathbf{X} \in \mathbb{R}^{m \times p_1}$ and $\mathbf{Y} \in \mathbb{R}^{m \times p_2}$, having p_1 and p_2 neurons respectively. We use the Gram matrices $\mathbf{K} = \mathbf{X}\mathbf{X}^\top$ and $\mathbf{L} = \mathbf{Y}\mathbf{Y}^\top$ to compute CKA:

$$\mathrm{CKA}(\mathbf{K}, \mathbf{L}) = \frac{\mathrm{HSIC}(\mathbf{K}, \mathbf{L})}{\sqrt{\mathrm{HSIC}(\mathbf{K}, \mathbf{K})\mathrm{HSIC}(\mathbf{L}, \mathbf{L})}}, \qquad (1)$$

where HSIC is the Hilbert-Schmidt independence criterion [Gretton *et al.*, 2007]. Given the centering matrix $\mathbf{H} = \mathbf{I}_n - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$, $\mathbf{K}' = \mathbf{H}\mathbf{K}\mathbf{H}$ and $\mathbf{L}' = \mathbf{H}\mathbf{L}\mathbf{H}$ are centered Gram matrices, then we have $\mathrm{HSIC}(\mathbf{K}, \mathbf{L}) = \mathrm{vec}(\mathbf{K}') \cdot \mathrm{vec}(\mathbf{L}')/(m-1)^2$. Thanks to the properties of CKA, invariant to orthogonal transformation and isotropic scaling, we are able to conduct a meaningful analysis of neural network representations. However, naive computation of CKA requires maintaining the activations across the entire dataset in memory, causing much memory consumption. To avoid this, we use minibatch estimators of CKA[Nguyen *et al.*, 2020], with a minibatch of 300 by iterating over the test dataset 10 times.

3.3 Representation Structure of EDT

We begin our investigation by studying the internal representation structure of our models. How are representations propagated within models in different low-level tasks? To answer this intriguing question, we compute CKA similarities between every pair of layers within a model. Apart from the convolutional head and tail, we include outputs of attention and FFN after residual connections in the transformer body.

We observe a block-diagonal structure in the CKA similarity maps in Fig. 2. As for the SR and deraining models in Fig. 2 (a)-(b), we find there are roughly four groups, among which a range of transformer layers are of high similarity. The first and last group structures (from left to right) correspond to the model head and tail, while the second and third group structures account for the transformer body. As for the denoising task (Fig. 2 (c)), there are only three obvious group structures, where the second one (transformer body) is dominated. Finally, from the cross-model comparison in Fig. 2 (d) and (h), we find *higher similarity* scores between denoising body layers and the second group SR layers, while showing *significant differences* compared to the third group SR layers.

We also explore the impact of single-task pre-training on the internal representations. As for SR and deraining in Fig. 2 (e)-(f), the representations of the model head and tail remain basically unchanged. Meanwhile, we observe *obvious representation changes* in the transition regions between the second and third groups. In terms of denoising in Fig. 2 (g), the internal representations do not change too much, consistent with the finding in Table 4 that denoising tasks obtain fewer improvements, compared to SR and deraining tasks.

Key Findings: (1) SR and deraining models show clear stages in the internal representations of the transformer body, while the denoising model presents a relatively uniform structure; (2) the denoising model layers show more similarity to the lower layers of SR models, containing more local information, as verified in Sec. 3.4; (3) single-task pre-training mainly affects the intermediate layers of SR and deraining models but has limited impact on the denoising model.

3.4 Single- and Multi-Task Pre-training

In the previous section, we observe that the transformer body of SR models is clearly composed of two group structures and pre-training mainly changes the representations of higher layers. What is the difference between these two partitions?

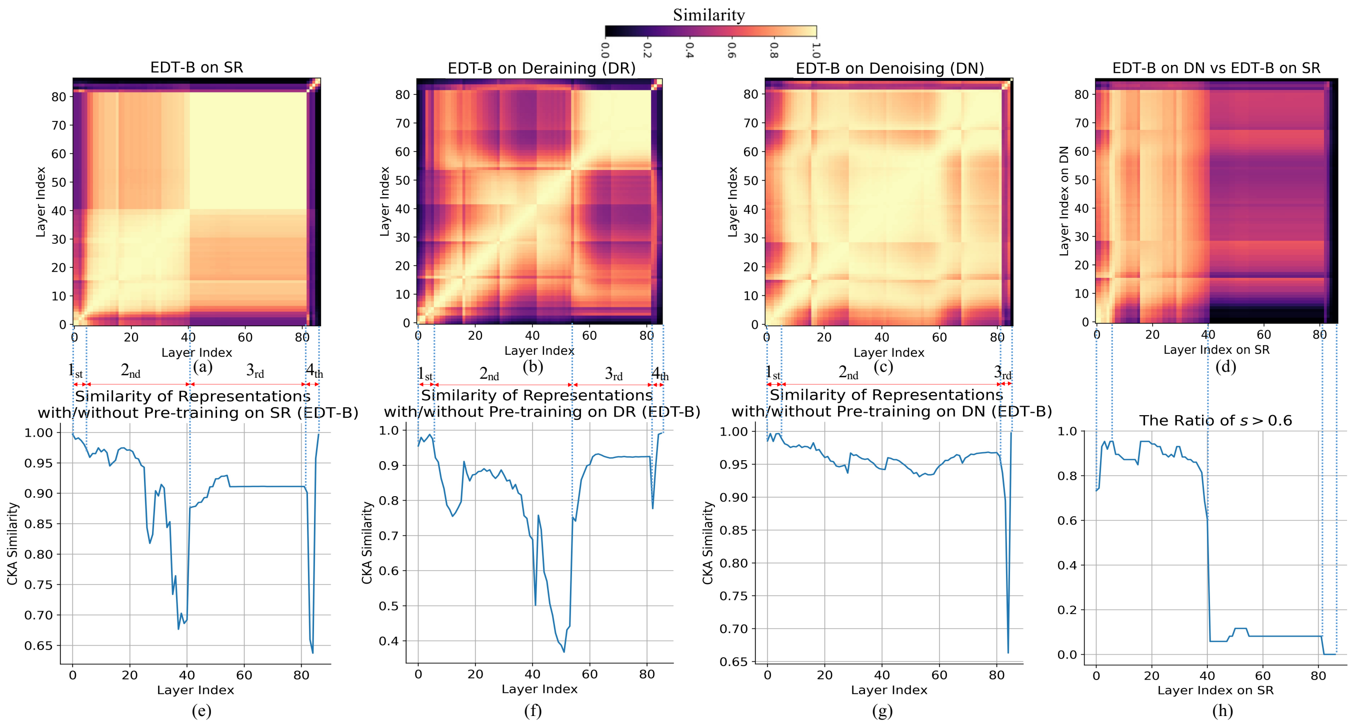

Figure 2: Sub-figures (a)-(c) show CKA similarities between all pairs of layers in $\times 2$ SR, light streak deraining and level-15 denoising EDT-B models with single-task pre-training, and the corresponding similarities between *with* and *without* pre-training are shown in (e)-(g). Sub-figure (d) shows the cross-model comparison between SR and denoising models and (h) shows the ratios of layer similarity larger than 0.6 for input images, where "s" means the similarity between the current layer in SR and any layer in denoising.

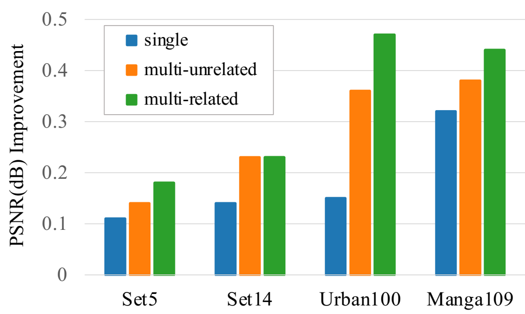

Figure 3: PSNR improvements of single-task, multi-unrelated-task and multi-related-task pre-training for EDT-B in $\times 2$ SR.

How does the pre-training, especially multi-task pre-training, affect the behaviors of models?

We conjecture that one possible reason causing the partition lies with the difference of ability to incorporate local or global information between different layers. We start by analyzing self-attention layers for their mechanism of dynamically aggregating information from other spatial locations, which is quite different from the fixed receptive field of the FFN layer. To represent the range of attentive fields, we average pixel distances between the queries and keys using attention weights for each head over 170,000 data points, where a larger distance usually refers to using more global information. We do not record attention distances of shifted local windows, because the shift operation narrows down boundary windows and hence can not reflect real distances.

As shown in Fig. 4 (e)-(h), for the second group structure (counted from the head, same as Sec. 3.3), the standard deviation of attention distances (shown as the blue area) is large and the mean value is small, indicating the attention modules in this group structure area have a mix of local heads (relatively small distances) and global heads (relatively large distances). On the contrary, the third group structure only contains global heads, showing more global information are aggregated in this stage.

Compared to single-task pre-training ($\times 2$ SR, Fig. 4 (b) and (f)), multi-unrelated-task setup ($\times 2$, $\times 3$ SR, g15 denoising, in Fig. 4 (c) and (g)) converts more global representations (in red box) of the third group to local ones, increasing the scope of the second group. In consequence, as shown in Fig. 3, we observe obvious PSNR improvements on all benchmarks. When replacing the g15 denoising with highly related $\times 4$ SR ($\times 2$, $\times 3$, $\times 4$ SR, in Fig. 4 (d) and (h)), we observe more changes in global representations, along with further improvements in Fig. 3. The inferiority of multi-unrelated-task setup is mainly due to the representation mismatch of unrelated tasks, as shown in Sec. 3.3. We also provide detailed quantitative comparisons for all tasks and different batch size settings in the supplementary material.

Key Findings: (1) the representations of SR models contain more local information in early layers while more global information in higher layers; (2) all three pre-training methods can greatly improve the performance by introducing different degrees of local information, treated as a kind of inductive bias, to the intermediate layers of the model, among which *multi-related-task pre-training performs best*.

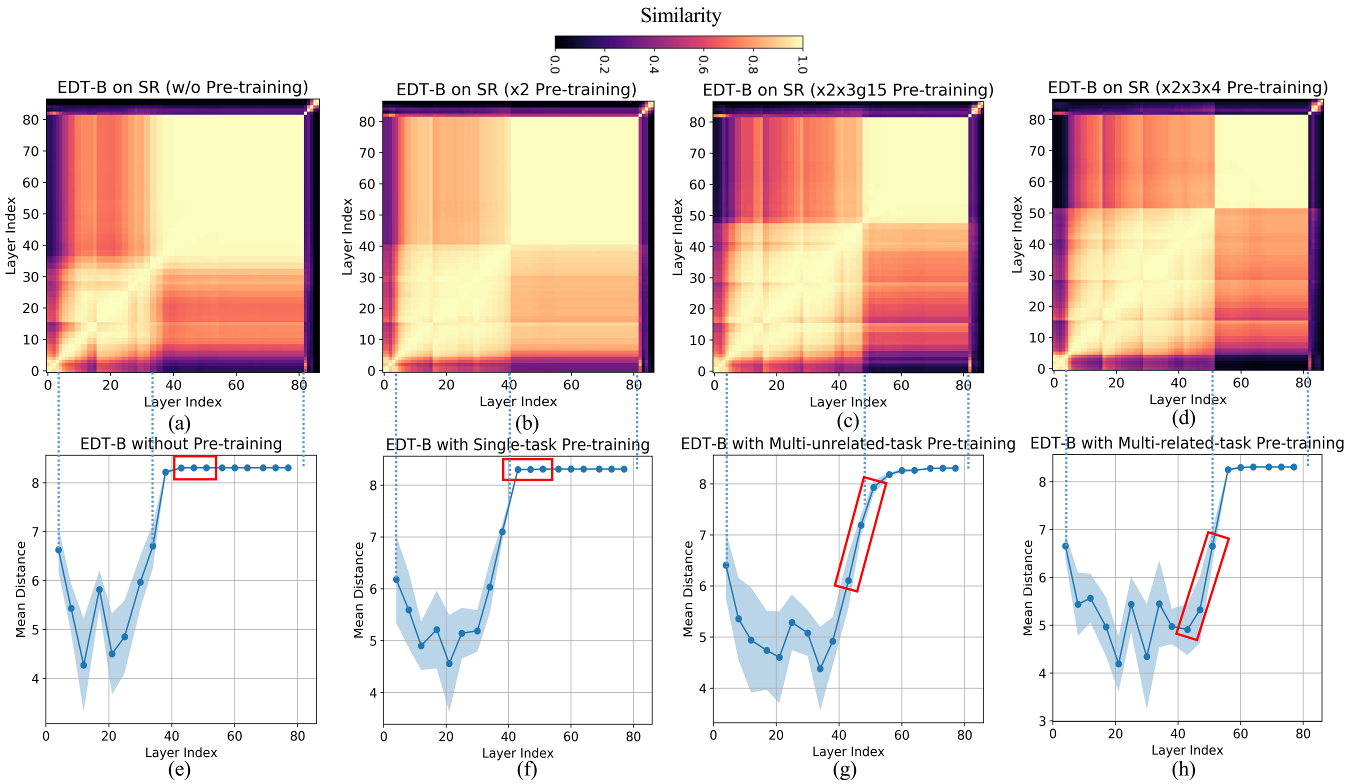

Figure 4: Sub-figures (a)-(d) show CKA similarities of $\times 2$ SR models, without pre-training as well as with pre-training on *a single task* ($\times 2$), *unrelated tasks* ($\times 2$, $\times 3$ SR, g15 denoising) and *highly related tasks* ($\times 2$, $\times 3$, $\times 4$ SR). Sub-figures (e)-(h) show the corresponding attention head mean distances of transformer blocks. We do not plot shifted local windows in (e)-(h) so that the last blue dotted line ("---") has no matching point. The red boxes indicate the same attention modules.

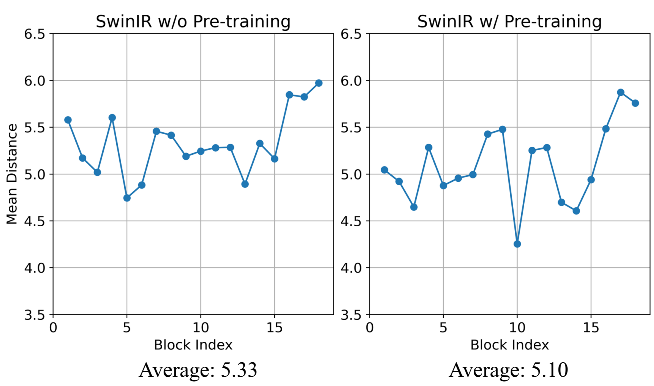

Figure 5: Attention head mean distances of transformer blocks in SwinIR with and without pre-training.

To validate whether the finding that pre-training brings more local information to the model also fit other window-based frameworks, we show the attention head distances of SwinIR [Liang *et al.*, 2021] in Fig. 5. Without pre-training, the first few blocks (1-15) tend to be local while the last ones (16-18) are more global. And pre-training brings more local representations, matching our observation before.

3.5 Effect of Data Scale on Pre-training

In this section, we investigate how pre-training data scale affects the super-resolution performance. As shown in Table 2,

Model	Data	Set5	Set14	Urban100	Manga109
EDT-B	0	38.45	34.57	33.80	39.93
EDT-B†	50K	38.53	34.66	33.86	40.14
EDT-B†	100K	38.55	34.68	33.90	40.18
EDT-B†	200K	38.56	34.71	33.95	40.25
EDT-B†	400K	38.61	34.75	34.05	**40.37**
EDT-B*	200K	**38.63**	**34.80**	**34.27**	**40.37**

Table 2: PSNR(dB) results of different pre-training data scales in $\times 2$ SR. "EDT-B†" refers to the base model with single-task ($\times 2$ SR) pre-training and "EDT-B*" represents the base model with multi-related-task ($\times 2$, $\times 3$, $\times 4$ SR) pre-training.

with regard to the EDT-B model, we obviously observe incremental PSNR improvements on multiple SR benchmarks by increasing the data scale from 50K to 400K during single-task pre-training. It is noted that we double the pre-training iterations for the data scale of 400K so that the data can be fully functional. However, a longer pre-training period largely increases the training burden.

On the contrary, as shown in Table 2, multi-related-task pre-training (with much fewer training iterations) successfully breaks through the limit. Our EDT-B model with multi-related-task pre-training on 200K images achieves new state of the arts on all benchmarks, though a smaller data scale is adopted, revealing that simply increasing the data scale may not be the optimal option. Thus, we suggest multi-related-task pre-training is more effective and data-efficient.

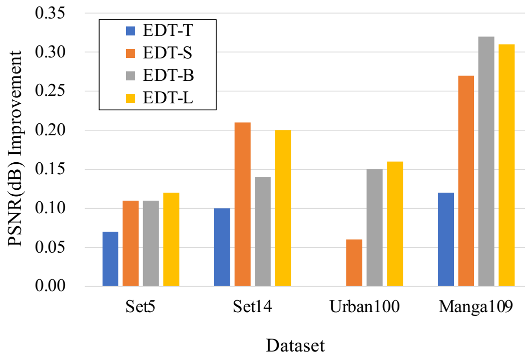

Figure 6: PSNR improvements of four variants of EDT models using single-task pre-training in $\times 2$ SR. "T", "S", "B" and "L" refer to tiny, small, base and large models. The improvement of EDT-T on Urban100 is 0.00dB, thus we do not plot the bar.

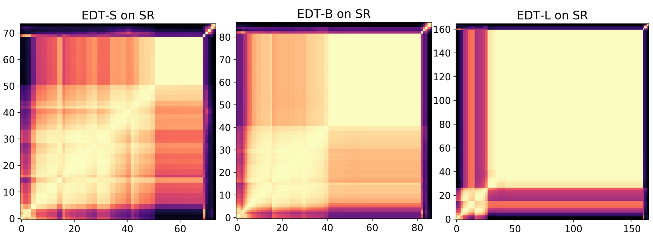

Figure 7: CKA similarities between all pairs of layers in EDT-S, EDT-B and EDT-L models using single-task pre-training in $\times 2$ SR.

3.6 Effect of Model Size on Pre-training

We conduct experiments to compare the performance of single-task pre-training for four model variants in the $\times 2$ SR task. As shown in Fig. 6, we visualize PSNR improvements of models with pre-training over counterparts trained from scratch. It is observed that models with larger capacities generally obtain more improvements. Especially, we find pre-training can still improve a lot upon already strong EDT-L models, showing the potential of pre-training. The quantitative results are provided in the supplementary file.

Here we visualize the CKA maps of the EDT-S, EDT-B and EDT-L models in Fig 7. As illustrated in Sec. 3.3, we already know there are roughly four group structures in the CKA maps of SR models, among which the second and third group structures account for the transformer body. The proportion of the third part is positively correlated with the model size. Especially, compared to the other two, the third group structure of EDT-L account for the vast majority and show high similarities, which reflects the redundancy of the model.

3.7 EDT v.s. ConvNets with Pre-training

We further explore the pre-training performance of EDT and CNNs-based models (RRDB [Wang *et al.*, 2018] and RCAN [Zhang *et al.*, 2018b]). Fig. 8 demonstrates that our EDT-B obtains greater or comparable improvements from pre-training, giving higher baselines with fewer parameters. From the representation comparisons between EDT and CNNs-based models exhibited in the supplementary material, we argue that the superiority of transformers may come from the utilization of global information.

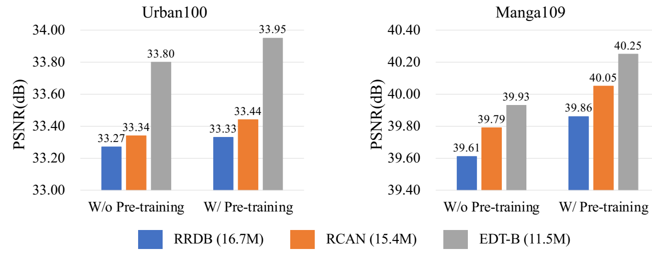

Figure 8: Quantitative comparison between ConvNets (RRDB and RCAN) and our EDT-B without ("W/o") and with ("W/") single-task pre-training in $\times 2$ SR.

4 Experiments

Following the pre-training guidelines, we conduct experiments in super-resolution (SR), denoising and detraining. As aforementioned, we observe that multi-related-task pre-training is highly effective and data-efficient. Thus, we adopt this pre-training strategy in all the tests. The involved pre-training tasks of SR include $\times 2$, $\times 3$ and $\times 4$, those of denoising include g15, g25 and g50, and those of deraining include light and heavy rain streaks. More experimental settings and visual comparisons are given in the supplementary file.

4.1 Super-Resolution Results

For the super-resolution (SR) task, we test our models on two settings, classical and lightweight SR, where the latter generally refers to models with $<$ 1M parameters. The results of $\times 3$ classical SR and lightweight SR are provided in the supplementary material due to the space limit.

We compare our EDT with state-of-the-art CNNs-based methods as well as transformer-based methods. As shown in Table 3, while the proposed EDT-B serves as a strong baseline, achieving nearly 0.1dB gains on multiple datasets over SwinIR [Liang *et al.*, 2021], pre-training still brings significant improvements on $\times 2$ and $\times 4$ scales. For example, we observe up to *0.46*dB and *0.45*dB improvements on high-resolution benchmark Urban100 and Manga109, manifesting the effectiveness of our pre-training strategy.

4.2 Denoising Results

In Table 4, we present our three models: (1) EDT-B without pre-training; (2) EDT-B with pre-training; (3) EDT-B without downsampling and pre-training.

It is worthwhile to note that, unlike SR models that benefit a lot from pre-training, denoising models only achieve 0.02-0.11dB gains. One possible reason is that we use a large training dataset in denoising tasks, which already provides sufficient data to make the capacity of our models into full play. On the other hand, pre-training hardly affects the internal feature representation of models, discussed in Sec. 3.3. Therefore, we suggest that the Gaussian denoising task may not need a large amount of training data.

Besides, we find our framework is well performed on high noise levels (e.g., $\sigma = 50$), while yielding slightly inferior performance on low noise levels (e.g., $\sigma = 15$). This could be caused by the downsampling operation in EDT. To verify this assumption, we train another EDT-B model without downsampling. As shown in Table 4, it does obtain better

Scale	Method	#Param. ($\times 10^6$)	Set5		Set14		BSDS100		Urban100		Manga109	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
$\times 2$	RCAN [Zhang *et al.*, 2018b]	15.4	38.27	0.9614	34.12	0.9216	32.41	0.9027	33.34	0.9384	39.44	0.9786
	SAN [Dai *et al.*, 2019]	15.7	38.31	0.9620	34.07	0.9213	32.42	0.9028	33.10	0.9370	39.32	0.9792
	NLSA [Mei *et al.*, 2021]	31.9	38.34	0.9618	34.08	0.9231	32.43	0.9027	33.42	0.9394	39.59	0.9789
	IPT† [Chen *et al.*, 2021]	115.5	38.37	-	34.43	-	32.48	-	33.76	-	-	-
	SwinIR [Liang *et al.*, 2021]	11.8	38.42	0.9622	34.48	0.9252	32.50	0.9038	33.70	0.9418	39.81	0.9796
	SwinIR‡ [Liang *et al.*, 2021]	11.8	38.42	0.9623	34.46	0.9250	32.53	0.9041	33.81	0.9427	39.92	0.9797
	EDT-B(Ours)	11.5	38.45	0.9624	34.57	0.9258	32.52	0.9041	33.80	0.9425	39.93	0.9800
	EDT-B†(Ours)	11.5	38.63	0.9632	34.80	0.9273	32.62	0.9052	34.27	0.9456	40.37	0.9811
$\times 4$	RCAN [Zhang *et al.*, 2018b]	15.6	32.63	0.9002	28.87	0.7889	27.77	0.7436	26.82	0.8087	31.22	0.9173
	SAN [Dai *et al.*, 2019]	15.9	32.64	0.9003	28.92	0.7888	27.78	0.7436	26.79	0.8068	31.18	0.9169
	NLSA [Mei *et al.*, 2021]	44.2	32.59	0.9000	28.87	0.7891	27.78	0.7444	26.96	0.8109	31.27	0.9184
	IPT† [Chen *et al.*, 2021]	115.6	32.64	-	29.01	-	27.82	-	27.26	-	-	-
	SwinIR [Liang *et al.*, 2021]	11.9	32.74	0.9020	29.06	0.7939	27.89	0.7479	27.37	0.8233	31.93	0.9246
	SwinIR‡ [Liang *et al.*, 2021]	11.9	32.92	0.9044	29.09	0.7950	27.92	0.7489	27.45	0.8254	32.03	0.9260
	EDT-B(Ours)	11.6	32.82	0.9031	29.09	0.7939	27.91	0.7483	27.46	0.8246	32.05	0.9254
	EDT-B†(Ours)	11.6	33.06	0.9055	29.23	0.7971	27.99	0.7510	27.75	0.8317	32.39	0.9283

Table 3: Quantitative comparison for classical SR on PSNR(dB)/SSIM on the Y channel from the YCbCr space. "‡" means the $\times 4$ model of SwinIR are pre-trained on the $\times 2$ setup and training patch size is 64×64 (ours is 48×48). "†" indicates methods with a pre-training. Best and second best results are in red and blue colors.

Dataset	σ	BM3D [Dabov *et al.*, 2007]	DnCNN [Zhang *et al.*, 2017]	FFDNet [Zhang *et al.*, 2018a]	BRDNet [Tian *et al.*, 2020]	IPT† [Chen *et al.*, 2021]	DRUNet [Zhang *et al.*, 2021]	SwinIR‡ [Liang *et al.*, 2021]	**EDT-B** (Ours)	**EDT-B**† (Ours)	**EDT-B*** (Ours)
CBSD68	15	33.52	33.90	33.87	34.10	-	34.30	34.42	34.33	34.38	34.39
	25	30.71	31.24	31.21	31.43	-	31.69	31.78	31.73	31.76	31.76
	50	27.38	27.95	27.96	28.16	28.39	28.51	28.56	28.55	28.57	28.56
Kodak24	15	34.28	34.60	34.63	34.88	-	35.31	35.34	35.25	35.31	35.37
	25	32.15	32.14	32.13	32.41	-	32.89	32.89	32.84	32.89	32.94
	50	28.46	28.95	28.98	29.22	29.64	29.86	29.79	29.81	29.83	29.87
McMaster	15	34.06	33.45	34.66	35.08	-	35.40	35.61	35.43	35.51	35.61
	25	31.66	31.52	32.35	32.75	-	33.14	33.20	33.20	33.26	33.34
	50	28.51	28.62	29.18	29.52	29.98	30.08	30.22	30.21	30.25	30.25
Urban100	15	33.93	32.98	33.83	34.42	-	34.81	35.13	34.93	35.04	35.22
	25	31.36	30.81	31.40	31.99	-	32.60	32.90	32.78	32.86	33.07
	50	27.93	27.59	28.05	28.56	29.71	29.61	29.82	29.93	29.98	30.16

Table 4: Quantitative comparison for color image denoising on PSNR(dB) on RGB channels. "‡" means the $\sigma = 25/50$ models of SwinIR are pre-trained on the $\sigma = 15$ level. "†" indicates methods with pre-training. "$*$" means our model *without pre-training* and downsampling.

performance on the low level noises. Nonetheless, we suggest that the proposed EDT model is still a good choice for denoising tasks since *it strikes a sweet point between performance and computational complexity*. For example, the FLOPs of EDT-B (38G) is only *8.4%* of SwinIR (451G).

4.3 Deraining Results

We evaluate the performance of our EDT on Rain100L [Yang *et al.*, 2019] and Rain100H [Yang *et al.*, 2019] two datasets, accounting for light and heavy rain streaks. As shown in Table 5, though the model size of our EDT-B (11.5M) for deraining is far smaller than IPT (116M), it still outperforms IPT by *0.52*dB on the light rain setting. Meanwhile, our model reaches significantly superior results by *2.66*dB gain on the heavy rain setting, compared to the second-best RCDNet [Wang *et al.*, 2020], supporting that EDT performs well for restoration tasks with heavy degradation.

Method	RAIN100L		RAIN100H	
	PSRN	SSIM	PSRN	SSIM
DSC [Luo *et al.*, 2015]	27.34	0.8494	13.77	0.3199
GMM [Li *et al.*, 2016]	29.05	0.8717	15.23	0.4498
JCAS [Gu *et al.*, 2017]	28.54	0.8524	14.62	0.4510
Clear [Fu *et al.*, 2017a]	30.24	0.9344	15.33	0.7421
DDN [Fu *et al.*, 2017b]	32.38	0.9258	22.85	0.7250
RESCAN [Li *et al.*, 2018]	38.52	0.9812	29.62	0.8720
PReNet [Ren *et al.*, 2019]	37.45	0.9790	30.11	0.9053
SPANet [Wang *et al.*, 2019]	35.33	0.9694	25.11	0.8332
JORDER_E [Yang *et al.*, 2019]	38.59	0.9834	30.50	0.8967
SSIR [Wei *et al.*, 2019]	32.37	0.9258	22.47	0.7164
RCDNet [Wang *et al.*, 2020]	40.00	0.9860	31.28	0.9093
IPT† [Chen *et al.*, 2021]	41.62	0.9880	-	-
EDT-B†(Ours)	42.14	0.9903	34.02	0.9406

Table 5: PSNR(dB)/SSIM results for image deraining on the Y channel. "†" indicates methods with pre-training.

5 Conclusion

Based on the proposed framework, we perform an in-depth analysis of transformer-based image pre-training in low-level vision. We find pre-training plays the central role of developing stronger intermediate representations by incorporating more local information. Also, we find the effect of pre-training is task-specific, leading to significant improvements on SR and deraining while limited gains on denoising. Then, we suggest multi-related-task pre-training exhibits great potential in digging image priors, far more efficient than using larger pre-training datasets. Finally, we show how data scale and model size affect the performance of pre-training and present comparisons between transformers and ConvNets.

Acknowledgments

This work is partially supported by Shenzhen Science and Technology Program KQTD20210811090149095 and also the Pearl River Talent Recruitment Program 2019QN01X226.

References

[Agustsson and Timofte, 2017] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *CVPRW*, pages 126–135, 2017.

[Brown *et al.*, 2020] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[Chen *et al.*, 2017] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *PAMI*, 40(4):834–848, 2017.

[Chen *et al.*, 2021] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *CVPR*, pages 12299–12310, 2021.

[Chen *et al.*, 2022] X Chen, X Wang, J Zhou, and C Dong. Activating more pixels in image super-resolution transformer. arxiv 2022. *arXiv preprint arXiv:2205.04437*, 2022.

[Cortes *et al.*, 2012] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, 13(1):795–828, 2012.

[Dabov *et al.*, 2007] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *TIP*, 16(8):2080–2095, 2007.

[Dai *et al.*, 2019] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *CVPR*, pages 11065–11074, 2019.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Dosovitskiy *et al.*, 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020.

[Fu *et al.*, 2017a] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley. Clearing the skies: A deep network architecture for single-image rain removal. *TIP*, 26(6):2944–2956, 2017.

[Fu *et al.*, 2017b] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *CVPR*, pages 3855–3863, 2017.

[Girshick *et al.*, 2014] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[Girshick, 2015] Ross Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015.

[Gretton *et al.*, 2007] Arthur Gretton, Kenji Fukumizu, Choon Hui Teo, Le Song, Bernhard Schölkopf, Alexander J Smola, et al. A kernel statistical test of independence. In *Nips*, volume 20, pages 585–592. Citeseer, 2007.

[Gu *et al.*, 2017] Shuhang Gu, Deyu Meng, Wangmeng Zuo, and Lei Zhang. Joint convolutional analysis and synthesis sparse representation for single image layer separation. In *ICCV*, pages 1708–1716, 2017.

[He *et al.*, 2021] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[Huang *et al.*, 2015] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, pages 5197–5206, 2015.

[Kolesnikov *et al.*, 2020] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *ECCV*, pages 491–507. Springer, 2020.

[Kornblith *et al.*, 2019] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *CVPR*, pages 2661–2671, 2019.

[LeCun *et al.*, 1989] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[Li *et al.*, 2016] Yu Li, Robby T Tan, Xiaojie Guo, Jiangbo Lu, and Michael S Brown. Rain streak removal using layer priors. In *CVPR*, pages 2736–2744, 2016.

[Li *et al.*, 2018] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *ECCV*, pages 254–269, 2018.

[Liang *et al.*, 2021] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *ICCVW*, pages 1833–1844, 2021.

[Liu *et al.*, 2021] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *ICCV*, 2021.

[Liu *et al.*, 2022] Lin Liu, Lingxi Xie, Xiaopeng Zhang, Shanxin Yuan, Xiangyu Chen, Wengang Zhou, Houqiang Li, and Qi Tian. Tape: Task-agnostic prior embedding for image restoration. In *ECCV*, pages 447–464. Springer, 2022.

[Luo *et al.*, 2015] Yu Luo, Yong Xu, and Hui Ji. Removing rain from a single image via discriminative sparse coding. In *ICCV*, pages 3397–3405, 2015.

[Mahajan *et al.*, 2018] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, pages 181–196, 2018.

[Mei *et al.*, 2021] Yiqun Mei, Yuchen Fan, and Yuqian Zhou. Image super-resolution with non-local sparse attention. In *CVPR*, pages 3517–3526, 2021.

[Nguyen *et al.*, 2020] Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. *arXiv preprint arXiv:2010.15327*, 2020.

[Radford *et al.*, 2018] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[Radford *et al.*, 2019] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[Raghu *et al.*, 2021] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *arXiv preprint arXiv:2108.08810*, 2021.

[Ren *et al.*, 2019] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: A better and simpler baseline. In *CVPR*, pages 3937–3946, 2019.

[Sun *et al.*, 2017] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, pages 843–852, 2017.

[Tian *et al.*, 2020] Chunwei Tian, Yong Xu, and Wangmeng Zuo. Image denoising using deep cnn with batch renormalization. *Neural Networks*, 121:461–473, 2020.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[Wang *et al.*, 2018] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *ECCVW*, pages 0–0, 2018.

[Wang *et al.*, 2019] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson WH Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *CVPR*, pages 12270–12279, 2019.

[Wang *et al.*, 2020] Hong Wang, Qi Xie, Qian Zhao, and Deyu Meng. A model-driven deep neural network for single image rain removal. In *CVPR*, pages 3103–3112, 2020.

[Wang *et al.*, 2021a] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021.

[Wang *et al.*, 2021b] Zhendong Wang, Xiaodong Cun, Jianmin Bao, and Jianzhuang Liu. Uformer: A general u-shaped transformer for image restoration. *arXiv preprint arXiv:2106.03106*, 2021.

[Wei *et al.*, 2019] Wei Wei, Deyu Meng, Qian Zhao, Zongben Xu, and Ying Wu. Semi-supervised transfer learning for image rain removal. In *CVPR*, pages 3877–3886, 2019.

[Xiao *et al.*, 2021] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *arXiv preprint arXiv:2106.14881*, 2021.

[Yang *et al.*, 2019] Wenhan Yang, Robby T Tan, Jiashi Feng, Zongming Guo, Shuicheng Yan, and Jiaying Liu. Joint rain detection and removal from a single image with contextualized deep networks. *PAMI*, 42(6):1377–1393, 2019.

[Zamir *et al.*, 2022] 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 *CVPR*, pages 5728–5739, 2022.

[Zhang *et al.*, 2017] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *TIP*, 26(7):3142–3155, 2017.

[Zhang *et al.*, 2018a] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *TIP*, 27(9):4608–4622, 2018.

[Zhang *et al.*, 2018b] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, pages 286–301, 2018.

[Zhang *et al.*, 2021] Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *PAMI*, 2021.