MULLER: Multilayer Laplacian Resizer for Vision

Zhengzhong Tu, Peyman Milanfar, Hossein Talebi
Google Research
zhengzhongtu@google.com

Abstract

Image resizing operation is a fundamental preprocessing module in modern computer vision. Throughout the deep learning revolution, researchers have overlooked the potential of alternative resizing methods beyond the commonly used resizers that are readily available, such as nearest-neighbors, bilinear, and bicubic. The key question of our interest is whether the front-end resizer affects the performance of deep vision models? In this paper, we present an extremely lightweight multilayer Laplacian resizer with only a handful of trainable parameters, dubbed MULLER resizer. MULLER has a bandpass nature in that it learns to boost details in certain frequency subbands that benefit the downstream recognition models. We show that MULLER can be easily plugged into various training pipelines, and it effectively boosts the performance of the underlying vision task with little to no extra cost. Specifically, we select a state-of-the-art vision Transformer, MaxViT [50], as the baseline, and show that, if trained with MULLER, MaxViT gains up to 0.6% top-1 accuracy, and meanwhile enjoys 36% inference cost saving to achieve similar top-1 accuracy on ImageNet-1k, as compared to the standard training scheme. Notably, MULLER's performance also scales with model size and training data size such as ImageNet-21k and JFT, and it is widely applicable to multiple vision tasks, including image classification, object detection and segmentation, as well as image quality assessment. The code is available at https://github.com/google-research/google-research/tree/master/muller.

1. Introduction

Most computer vision problems such as image classification, object detection, video recognition, and image/video generation have seen groundbreaking advancement by deep neural networks-based models that are trained on web-scale, human-curated datasets [11, 12, 15, 15, 24, 33, 35, 40, 41, 46, 53, 58]. In any of the underlining training infrastructures like Tensorflow [1] and PyTorch [30], image resizing

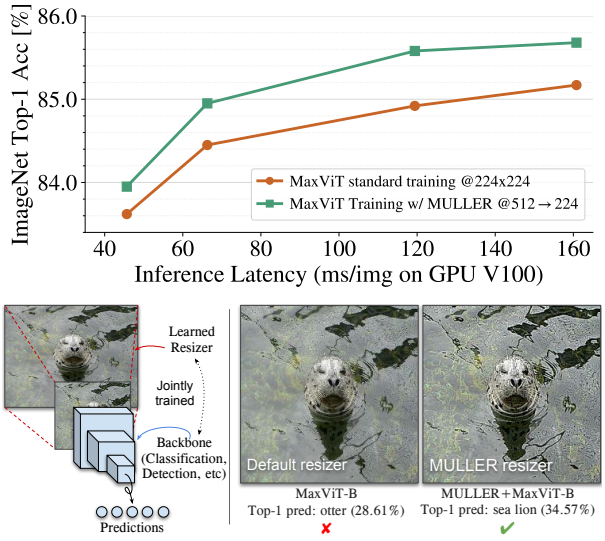


Figure 1. Top: our proposed learned resizer can push forward a strong vision Transformer MaxViT [50] by up to 0.6% top-1 accuracy on ImageNet-1K with no extra inference cost. Results for other backbones are shown in Sec. 4.2. Bottom: demonstratation of the learned resizer - with detail-boosted input image, the classification accuracy of the examplar image has improved.

is an essential preprocessing step which enables efficient gradient-based training of networks with millions of trainable parameters. Moreover, the factor of image size can sometimes significantly impact the performance of various tasks, particularly those requiring high-resolution prediction. Although neural architectures have been revolutionized by CNNs and Transformers, surprisingly limited attention has been paid to the role of image resizing operations.

Resizing or rescaling refers to the process of changing the resolution of an image, while largely preserving its content for human or machine perception. There are several major reasons for using resizing: (1) The mini-batch gradient-based training scheme requires the same image resolution in a batch, (2) Resizing can help to reduce computational complexity, making it easier and faster to train

and inference neural networks, (3) Smaller images consume lower memory footprint, enabling stable training of large models like Transformers with larger batch-size, (4) Resizing contributes to improving model generalization and robustness by reducing overfitting to specific image size and scales, making the models more flexible and applicable to real-world scenarios.

Moreover, resizing is an integral component of remote inference frameworks. Typically, to maintain the bandwidth efficiency of the communication network, before sending an image to the inference server, a thumbnail generator downscales the image to a fixed resolution (*e.g.* 480p). The thumbnail generator can be located on the client side (*e.g.* smart phone), or it can be part of a cloud storage system. This means that in most cases the inference server does not have access to the original image.

Basic resizing functions such as nearest-neighbor or bilinear interpolation have long been the go-to options with little to no deliberate consideration in most training software. While these simple methods offer greater simplicity and efficiency, they are not optimized for specific computer vision tasks and may lead to the loss of important visual features or details, which can, sometimes, result in significant performance degradation [29, 45]. To overcome this limitation, researchers have proposed learned resizers (or downsamplers) [3, 45] that leverage the deep neural networks to learn image resizing directly from data, yielding improved performance on several tasks. However, one of the main challenges with these learned resizers is that they often require a large number of parameters, and high computational overhead during training and inference. Note that this is specifically a bottleneck in remote inference where the resizer (a.k.a thumbnail generator) is not in the inference server, and may have limited computational resources to run a heavy neural net resizer. Additionally, less-bounded resizers can sometimes be difficult to transfer to new tasks or datasets due to their excessive model capability.

In this paper, we introduce an incredibly lightweight learned resizer, which we call MULLER, that operates on multilayer Laplacian decomposition of images (see Fig. 2). Our method requires very few parameters and FLOPs, and does not incur any extra training cost, outperforming existing methods in terms of computational efficiency, parameter efficiency, and transferability. We show that it is the ability-to-learn that makes a better resizer, but not the capacity of the resizer – our MULLER resizer only learns four parameters and is more effective than previous complex ones using deep residual blocks [45]. We also demonstrate that our method can be used as a drop-in replacement for off-the-shelf resizing functions on several vision tasks, including classification, object detection and segmentation, and image quality assessment, resulting in significant performance improvements without any extra cost. As shown in Fig. 1, for example, training with the MULLER resizer achieves up to 0.6% performance gain, using a state-of-the-art backbone MaxViT [50] as the testbed. Our contributions are:

- We propose a surprisingly simple and lightweight resizer, that can be used as a drop-in replacement for off-the-shelf resizing functions like bilinear resizing.
- We demonstrate its applications to multiple computer vision tasks, including image classification, object detection and segmentation, and image quality assessment, showing superior performance over existing approaches.
- Extensive ablation studies, analysis, and visualization results are provided to show the robustness and generalization of the proposed resizer for various model scales, benchmarks, and tasks.

2. Related Work

Resizing in vision. Resizing is a crucial preprocessing step to train deep learning vision models. Due to their simplicity, efficiency and availability, nearest-neighbor and bilinear interpolations are the most widely used resizing methods in both training, inference, and serving. These simple approaches, however, can suffer from detail loss and artifacts, and the degraded image quality might hamper the performance of downstream visual recognition tasks, especially when the resizing factor is large.

Some recent works have explored to use learning-based methods for image downscaling to enhance the desired content in the resized images from training data [3, 4, 20, 32, 45, 61]. For example, the authors of [3, 4] proposed a residual CNN module for downscaling, and jointly trained it with an image compression network to generate "compression-friendly" representations. [45] introduced a CNN-based learned resizer for various computer vision tasks, including image classification and image quality assessment. Similarly, the idea of learned rescaling has been applied to other computer vision applications [20, 32, 57, 61], showing improved performance in detection and recognition.

Image processing for machine vision. Image processing or enhancement problems such as super-resolution [19, 37], denoising [59], and deblurring [48, 49] have been long-standing challenges in computer vision. Recent works have focused on building larger-scale, diverse benchmarks, exploring novel model architectures, and improving training techniques. These works aim to produce visually pleasing outputs, often measured by conventional metrics like PSNR or SSIM [54], or through human evaluations, without considering downstream recognition performance of the output.

There exists a number of works relating image processing to image recognition performance, or machine-oriented image processing. Some works [17, 18, 34, 60] use image recognition accuracy as supplementary metrics besides

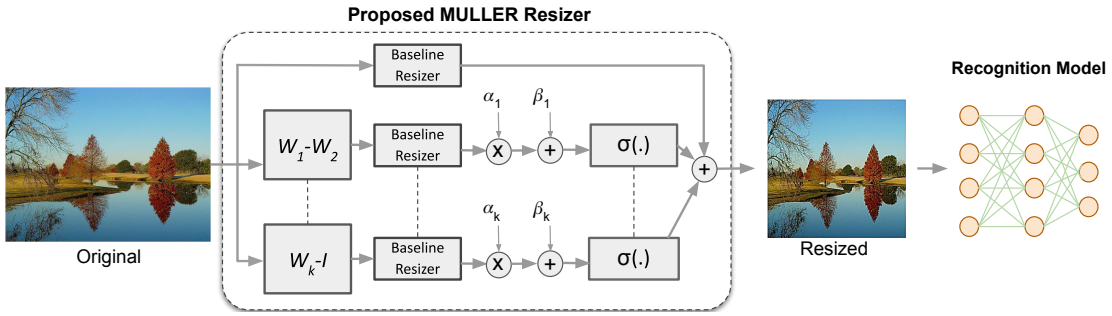


Figure 2. **The architecture of the proposed multilayer Laplacian resizer (MULLER).** The resizer decomposes the input image into multiple layers of Laplacian residuals and then adds them back to the default resized image. The MULLER resizer is jointly trained with the downstream recognition model.

visual quality metrics to evaluate the performance of image restoration models. Towards recognition-aware training, [10] proposed to train super-resolution using object detection loss and show promising results over conventional methods. Other works [39, 42] introduce pre-editing networks before image compression to improve compression efficiency without sacrificing classification accuracy. Recently, Liu et al. [26] developed an approach to train the processing models under the objective of image recognition accuracy, and investigated the efficacy of popular preprocessing operations such as super-resolution, denoising, and JPEG-deblocking on improving recognition performance. More similar studies [16, 22, 23, 36] have been conducted to jointly train a processing model like denoising, dehazing, face reconstruction, together with recognition model to achieve better image processing for recognition quality.

3. Proposed Approach

In this section, we introduce our proposed multilayer Laplacian resizer (MULLER, overviewed in Fig. 2), and discuss how we employ it for training several popular vision tasks. Unlike previous proposed resizers [3, 45], we aim to keep the computational cost of the model as low as possible such that it can replace existing resizers (e.g., bilinear) without extra cost, but also there is a notable performance gain. Our proposed approach is different in that (1) it is orders of magnitude faster, hence more scalable (to large image size), (2) it only has a handful of parameters which allows for better generalization, (3) it adds almost no extra training cost to the system. We show that with learning merely a couple of parameters, training with an added resizing module performs as effective as having a heavy downscaling network with several thousand parameter counts.

3.1. Resizer Model

Image resizing models can be generally formulated as:

$$\mathbf{y} = \mathbf{F}_2(\mathbf{R}(\mathbf{F}_1(\mathbf{x}); h', w')), \tag{1}$$

where $\mathbf{R}$ maps the input image $\mathbf{x}$ of size $h \times w$ to an output image of size $h' \times w'$ by computing the pixel values at the target spatial locations. $\mathbf{F}_1$ and $\mathbf{F}_2$ denote optional pre- and post-filtering operations. Typically, $\mathbf{F}_1$ and $\mathbf{F}_2$ can be identity functions, and $\mathbf{R}$ is chosen as a simple interpolation method like nearest-neighbor, bilinear, or bicubic. To learn more powerful resizing, learned resizers have been proposed [3, 4, 45] by applying a base resizer on intermediate neural activations, wherein $\mathbf{F}_1$ and $\mathbf{F}_2$ are two designed CNNs applied at the original and output resolutions, respectively. Despite showing promising performance, however, these resizers typically suffer from high computational complexity, and thus their net performance gain might be compromised in terms of the overall inference cost.

3.2. Proposed MULLER Resizer

We are inspired by the observations that these different learned resizers, if properly regularized, will often learn to enhance edges, details, or sharpness of the image to benefit downstream tasks [3, 4]. To this end, we present to date, the simplest learned resizing model, using multilayer Laplacian decomposition, that is able to achieve 'bandpassed' detail and texture manipulation with only a handful of learnable parameters. Fig. 2 shows the architecture of the proposed MULLER resizer. MULLER has the following form:

$$\mathbf{z} = \underbrace{\mathbf{R}(\mathbf{x})}_{\text{Base image}} + \sum_{\ell=1}^{k} \underbrace{\sigma(\alpha_\ell(\mathbf{R}((\mathbf{W}_\ell - \mathbf{W}_{\ell+1})\mathbf{x}) + \beta_\ell)}_{\text{Enhanced details in each subband}}, \tag{2}$$

where $\mathbf{R}$ denotes the base resizer (*e.g.* bilinear) and $\{\mathbf{W}_1, \mathbf{W}_2, ..., \mathbf{W}_k\}$ represents the low-pass filter basis. We define $\mathbf{W}_\ell$ as a positive row-stochastic matrix [27] of size $n \times n$, with n representing the number of pixels in the vectorized input image $\mathbf{x}$. Note that we assume $\mathbf{W}_{k+1} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Each layer in Eq. (2) (see Fig. 2) uses a difference of the filters to decompose the image into different detail layers (bandpass filtering).

Without loss of generality, we choose the Gaussian kernel as our base filter, and generate the filter bank by an iter-

ative application of the same base filter as $\mathbf{W}_\ell = \mathbf{W}^{k-\ell+1}$ with $\mathbf{W}$ being a Gaussian filter with standard deviation 1. Note that the iterative application of the low-pass filter results in a smoother image. The filtered subband image $(\mathbf{W}_\ell - \mathbf{W}_{\ell+1})\mathbf{x}$ in brach ℓ is fed into the base resizer to produce the target resolution layer. We add trainable scaling and bias parameters $(\alpha_\ell, \beta_\ell)$ per layer to modulate and shift the resized response. Then, a nonlinearity function σ (*e.g.*, `tanh`) is applied on the resulting image layer, and finally the output is added to the base resized image. Note that the scaling factor α_ℓ controls the amount of detail boosted or suppressed in layer l of the resizer, and the bias parameter β_ℓ controls the mean shift.

It is worth pointing out that in this framework, only the scalar and the bias values in the residual layers are trainable, meaning that for $k = 3$, there are only six trainable parameters, and the overall computational cost is only applying 4 bilinear resizers and 3 Gaussian filters. Note that the term "Laplacian" refers to an interpretation of the filtering structure in Fig. 2 that can be written as a summation of Laplacian operators, namely $\mathbf{L}_\ell = \mathbf{I} - \mathbf{W}_\ell$. More explicitly, for a linear activation, the resulting image $\mathbf{y}$ can be expressed as a Laplacian form [43]:

$$\mathbf{y} = \gamma_0\mathbf{R}(\mathbf{x}) + \gamma_1\mathbf{R}(\mathbf{L}_1\mathbf{x}) + ... + \gamma_k\mathbf{R}(\mathbf{L}_k\mathbf{x}) + \delta, \quad (3)$$

3.3. Applications in Vision Tasks

While theoretically the resizer can be a drop-in replacement of the default resizer anywhere in the data generation and machine learning pipeline, we mainly demonstrate its ability to learn more informative thumbnail images for downstream recognition and detection tasks, which account for most of the practical use cases. We showcase the impact of MULLER by jointly training it with the backbone, where the resizer takes a higher-resolution image from the data pipeline and downscales it to lower-size before feeding as the model inputs.[1] Since the proposed resizer is strongly regularized by its design, it needs no extra intermediate loss to train. The resizer is also task agnostic as no specific changes are needed to train with it in any framework on any vision or even vision-language tasks.

4. Experiments

We validate the performance of our proposed MULLER resizer on several competitive vision tasks on which resolution plays an important role on performance, including image classification, object detection and segmentation, and image quality assessment. In order to showcase the impact of MULLER, our main experiments include the state-of-the-art vision Transformer model MaxViT [50] as the baseline. We first demonstrate the performance of this baseline model by co-training it with MULLER. Then, we show that MULLER can be effective with other backbones such as ResNet [11], MobileNet-v2 [35] and EfficientNet-B0 [46]. In all the experiments, we use 2 layers in MULLER with Gaussian kernel size 5 and standard deviation 1. We use Tensorflow's default resizer as the base resizer. More experimental details can be found in supplementary materials.

[1]Note that MULLER is not limited to downscaling, and in fact should the original image data be low resolution, it can learn to upscale as well.

4.1. Main Experiments on ImageNet Classification

We demonstrate the efficacy of the MULLER resizer on the standard, but most competitive ImageNet-1K classification task [15]. We take a top-performing vision Transformer, MaxViT [50], as the backbone model, and pre-train it on ImageNet-1K at 224 × 224 resolution for 300 epochs. Instead of directly fine-tuning at higher resolution (*e.g.*, 384 or 512) like previous practices [8, 9, 24, 50], we jointly fine-tune the backbone with the MULLER resizer plugged before the stem layers. We set input and output resolutions as 512 and 224 for MULLER in the ImageNet experiments.

ImageNet-1K. The main results on ImageNet-1K classification are shown in Tab. 1 . Note that we include all the state-of-the-art models trained to their highest possible accuracy reported in the original papers. For better visualization, we draw the accuracy vs. FLOPs and accuracy vs. inference-latency scaling curves in Fig. 3, respectively. As may be seen, MaxViT powered by the MULLER resizer sets a new state-of-the-art top-1 accuracy 85.68% with only 43.9B FLOPs among all the compared models trained at 224x224. MULLER improves at an average of 0.49 accuracy across the four MaxViT variants. In terms of actual inference time, MaxViT with MULLER exceeds among all the models trained at various resolution – equivalently, it can save 36% latency to achieve $\sim 85.7\%$ accuracy.

ImageNet-21K and JFT. To demonstrate the scaling properties of the MULLER resizer with respect to data size, in Tab. 2 we report the results of the models pre-trained on ImageNet-21K and JFT-300M [38], respectively. It can be seen that with ImageNet-21k pretraining, the fine-tuned MaxViT with $\text{MULLER}_{512\to224}$ gains 0.8%, 0.6%, and 0.7% accuracy over directly finetuning without resizer for B, L, and XL models, respectively. Similarly for JFT-300M pretraining, those numbers are 0.8%, 0.7%, and 0.7%. It indicates that when finetuning with MULLER, MaxViT scales consistently when data size increases from ImageNet-1k up to JFT-300M.

We further observe that for larger models and larger training sets, the backbone can benefit even more through seeing larger input images. Thus, we also report the performance of training with $\text{MULLER}_{576\to288}$. We can see that it further boosts the performance by an average of 0.4∼0.5% across the board for both 21K and JFT. Remarkably, MaxViT-XL with $\text{MULLER}_{576\to288}$ achieves 89.16% top-1 accuracy with only 162.9B FLOPs.

Model	Eval size	Params	FLOPs	Thr (img/s)	IN-1K top-1 acc
•EffNetV2-S [47]	384	24M	8.8B	666.6	83.9
•EffNetV2-M [47]	480	55M	24.0B	280.7	85.1
•ConvNeXt-T [25]	224	29M	4.5B	774	82.1
•ConvNeXt-S [25]	224	50M	8.7B	447.1	83.1
•ConvNeXt-B [25]	224	89M	15.4B	292.1	83.8
•ConvNeXt-L [25]	224	198M	34.4B	146.8	84.3
○ViT-B/32 [9]	384	86M	55.4B	85.9	77.9
○ViT-B/16 [9]	384	307M	190.7B	27.3	76.5
○Swin-T [24]	224	29M	4.5B	755.2	81.3
○Swin-S [24]	224	50M	8.7B	436.9	83.0
○Swin-B [24]	224	88M	15.4B	278.1	83.5
○CSwin-B [8]	224	23M	4.3B	701	82.7
○CSwin-B [8]	224	35M	6.9B	437	83.6
○CSwin-B [8]	224	78M	15.0B	250	84.2
◇CoAtNet-0 [7]	224	25M	4.2B	526	81.6
◇CoAtNet-1 [7]	224	25M	4.2B	336	83.3
◇CoAtNet-2 [7]	224	75M	15.7B	247.7	84.1
◇CoAtNet-3 [7]	224	168M	34.7B	163.3	84.5
◇MaxViT-T	224	31M	5.6B	350.4	83.62
◇+MULLER$_{512\to224}$	224	31M	5.63B	349.6	83.95
◇MaxViT-S	224	69M	11.7B	242.5	84.45
◇+MULLER$_{512\to224}$	224	69M	11.73B	241.4	84.95
◇MaxViT-B	224	120M	23.4B	133.6	84.95
◇+MULLER$_{512\to224}$	224	120M	23.43B	133.0	85.58
◇MaxViT-L	224	212M	43.9B	99.4	85.17
◇+MULLER$_{512\to224}$	224	212M	43.93B	99.3	85.68

Table 1. **Performance comparison under the ImageNet-1K setting.** MULLER$_{A\to B}$ denote that MULLER resizes from A to B, where the backbone takes images of size B. FLOPs counts the total computation of the resizer and backbone. Throughput (Thr) is measured on a single V100 GPU with batch size 16, following [24, 25, 47]. •, ○, and ◇ denote ConvNets, Transformers, and hybrid models, respectively.

We also examine the generalization of the resizer across different model variants. We found that the learned weights in MULLER are very close across different variants, and the transferring results are as effective as the original training. Detailed results are in the supplementary materials. Note that we present our generalization experiments across different backbones in the next section.

4.2. Different Backbones

Main results. To explore the resizer beyond the MaxViT architecture, we selected some widely used backbones including ResNet-50 [11], EfficientNet-B0 [46], and MobileNet-v2 [35]. Our results are presented in Tab. 3. We also make comparisons with the resizer of Talebi et al. [45]. We observed that the proposed resizer improves the performance of the baseline backbones consistently. Also compared to

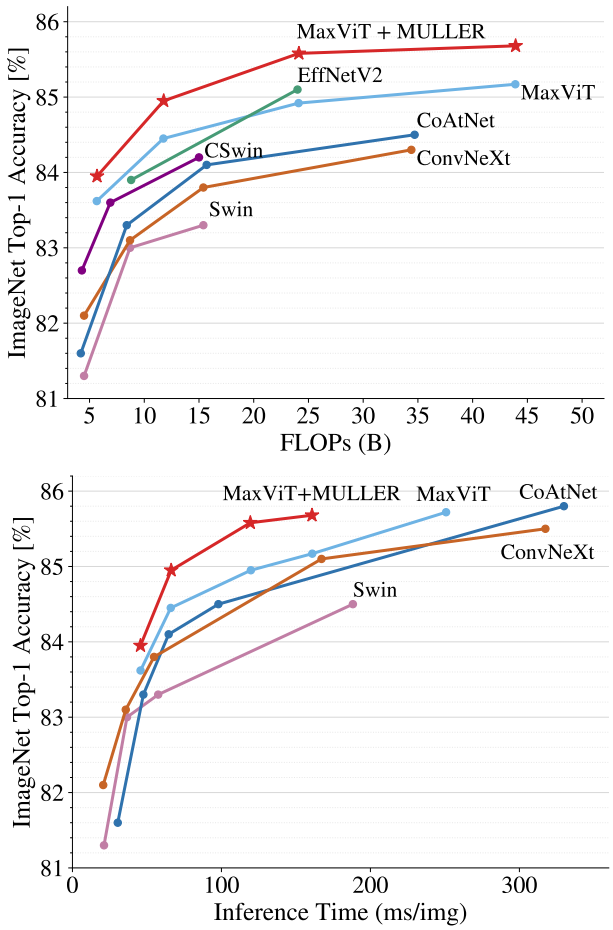


Figure 3. **Model FLOPs (top), and Inference Latency (bottom) performance comparison of state-of-the-art vision backbones on ImageNet-1K.** We show that MaxViT trained with MULLER resizer yielded the best accuracy *vs.* computation and accuracy *vs.* inference-cost tradeoff. Note that top figure includes only 224×224 models whereas the bottom figures include the best possible performance curves among various training size. Inference time is calculated by the throughput in Tab. 1.

[46], MULLER requires a significantly lower number of FLOPs (two orders-of-magnitude), and in some cases such as MobileNet-v2 and EfficientNet-B0 it outperforms [45]. These results also indicate that MULLER improves over baseline resizers low the FLOPs regime as well.

Cross-model Generalization. In order to examine the generalizability of MULLER, we evaluate classification models with resizers that are trained with other backbones. To this end, we first present the learned resizer parameters for each backbone, and then discuss the classification performances.

Results in Tab. 4 represent the learned MULLER parameters (see Eq. 2) for each backbone model trained on ImageNet-1k. We observed that (1) performance of the clas-

Model	Eval size	Params	FLOPs	IN-1K top-1 acc. 21K-pt	IN-1K top-1 acc. JFT-pt
•BiT-R-101x3 [14]	384	388M	204.6B	84.4	-
•BiT-R-152x4 [14]	480	937M	840.5B	85.4	-
•EffNetV2-L [47]	480	121M	53.0B	86.8	-
•EffNetV2-XL [47]	512	208M	94.0B	87.3	-
•ConvNeXt-L [25]	384	198M	101.0B	87.5	-
•ConvNeXt-XL [25]	384	350M	179.0B	87.8	-
•NFNet-F4+ [2]	512	527M	367B	-	89.20
○ViT-B/16 [9]	384	87M	55.5B	84.0	-
○ViT-L/16 [9]	384	305M	191.1B	85.2	
○ViT-L/16 [9]	512	305M	364B	-	87.76
○ViT-H/14 [9]	518	632M	1021B	-	88.55
○HaloNet-H4 [52]	512	85M	-	85.8	-
○SwinV2-B [24]	384	88M	-	87.1	-
○SwinV2-L [24]	384	197M	-	87.7	-
◇CvT-W24 [55]	384	277M	193.2B	87.7	-
◇R+ViT-L/16 [9]	384	330M	-	-	87.12
◇CoAtNet-3 [7]	384	168M	107.4B	87.6	88.52
◇CoAtNet-3 [7]	512	168M	214B	87.9	88.81
◇CoAtNet-4 [7]	512	275M	360.9B	88.1	89.11
◇MaxViT-B	224	119M	23.4B	86.63	87.05
◇+MULLER$_{512\to224}$	224	119M	23.4B	87.40	87.82
◇+MULLER$_{576\to288}$	288	119M	40.6B	87.92	88.39
◇MaxViT-L	224	212M	43.9B	86.86	87.72
◇+MULLER$_{512\to224}$	224	212M	43.9B	87.48	88.43
◇+MULLER$_{576\to288}$	288	212M	73.4B	87.94	88.87
◇MaxViT-XL	224	475M	97.8B	87.25	88.06
◇+MULLER$_{512\to224}$	224	475M	97.8B	87.90	88.74
◇+MULLER$_{576\to288}$	288	475M	162.9B	88.31	89.16
◇MaxViT-B	512	119M	138.3B	88.38	88.82
◇MaxViT-L	512	212M	245.2B	88.46	89.41
◇MaxViT-XL	512	475M	535.2B	88.70	89.53

Table 2. **Performance comparison for large-scale data regimes**: ImageNet-21K and JFT pretrained models. We report results using two different settings: MULLER$_{512\to224}$ and MULLER$_{576\to288}$ respectively, as we observe that on larger models and larger training sets, the backbone benefits more by seeing larger inputs.

sification models are more sensitive to α_1 than α_2, and (2) the learned bias values are relatively small, meaning the resizer does not significantly shift the mean of each residual image layer. Note that $|\alpha_\ell| > 1$ means the image details represented by the ℓ-th layer are boosted, whereas $|\alpha_\ell| < 1$ has the opposite effect.

To quantify generalizability of the resizer, we used the learned parameters in Tab. 4 to evaluate different backbones. As for different backbones, Tab. 5 shows that one model leads to classification performance that is in the average proximity of 0.15 from the best top-1 accuracy. We believe this can be explained by the fact that MULLER is a

Model	Size	FLOPs	top-1 acc.
EffNet-B0 [46]	224	0.39B	77.1
+ [45]$_{512\to224}$	224	2.63B	77.9
+MULLER$_{512\to224}$	224	0.42B	78.2
MobileNet-v2 [35]	224	0.60B	70.5
+ [45]$_{512\to224}$	224	2.84B	71.5
+MULLER$_{512\to224}$	224	0.63B	71.8
ResNet-50 [11]	224	6.97B	75.3
+ [45]$_{512\to224}$	224	9.33B	76.2
+MULLER$_{512\to224}$	224	7.0B	76.2

Table 3. **Performance comparison under ImageNet-1K setting with different backbones.**

Model	α_1	β_1	α_2	β_2
EffNet-B0 [46]	1.715	0.088	-8.41	0.001
MobileNet-v2 [35]	1.480	0.174	-5.25	-0.058
ResNet-50 [11]	1.892	-0.014	-11.295	0.003

Table 4. **The learned MULLER parameters for different backbone models train on ImageNet-1k.**

Model	EffNet-B0	MobileNet-v2	ResNet-50
MULLER$_{\text{EffNet}}$	78.2	71.6	75.9
MULLER$_{\text{MobileNet}}$	78.0	71.8	76.0
MULLER$_{\text{ResNet}}$	78.1	71.7	76.2

Table 5. **Cross-model validation of the MULLER resizer for ImageNet-1K on different backbones.**

constrained model with only 4 trainable parameters. Also, it is important to highlight that in contrast to the resizer in [45], MULLER does not require fine-tuning.

Impact of Aliasing. It has been shown that aliasing may impact the performance of some deep vision models [29,51]. It is worth mentioning that the results presented in this section are based on anti-aliased images. More specifically, we used the `AREA` downscaling method in TensorFlow to produce 512^2 inputs to MULLER. We observed that while removing anti-aliasing does not hamper the overall performance gain obtained by MULLER, the learned parameters may differ from Tab. 4. We will present our results without anti-aliasing in the supplementary material.

4.3. Downstream tasks

Object Detection and Instance Segmentation. We evaluated the performance of MULLER on COCO2017 [21] for object bounding box detection and instance segmentation tasks with a two-stage cascaded Mask-RCNN framework [31]. We warm start the MaxViT backbone using checkpoints pretrained on ImageNet-1K, then finetune the whole model including the resizer on COCO.

Tab. 6 summarizes the object detection and instance seg-

Backbone	Resolution	AP	AP_{50}	AP_{75}	AP^m	AP^m_{50}	AP^m_{75}	FLOPs
•ResNet-50 [11]	1280×800	46.3	64.3	50.5	40.1	61.7	43.4	739B
•X101-32 [56]	1280×800	48.1	66.5	52.4	41.6	63.9	45.2	819B
•X101-64 [56]	1280×800	48.3	66.4	52.3	41.7	64.0	45.1	972B
•ConvNeXt-T [25]	1280×800	50.4	69.1	54.8	43.7	66.5	47.3	741B
•ConvNeXt-S [25]	1280×800	51.9	70.8	56.5	45.0	68.4	49.1	827B
•ConvNeXt-B [25]	1280×800	52.7	71.3	57.2	45.6	68.9	49.5	964B
○Swin-T [24]	1280×800	50.4	69.2	54.7	43.7	66.6	47.3	745B
○Swin-S [24]	1280×800	51.9	70.7	56.3	45.0	68.2	48.8	838B
○Swin-B [24]	1280×800	51.9	70.5	56.4	45.0	68.1	48.9	982B
○UViT-T [6]	896×896	51.1	70.4	56.2	43.6	67.7	47.2	613B
○UViT-S [6]	896×896	51.4	70.8	56.2	44.1	68.2	48.0	744B
○UViT-B [6]	896×896	52.5	72.0	57.6	44.3	68.7	48.3	975B
◇MaxViT-T	640×640	49.9	69.9	54.6	42.7	66.6	46.4	379B
◇+$MULR_{896\rightarrow640}$	640×640	50.5	70.7	55.0	43.1	67.0	46.7	379B
◇MaxViT-S	640×640	50.5	70.2	55.3	43.3	67.3	46.8	432B
◇+$MULR_{896\rightarrow640}$	640×640	50.8	70.4	55.5	43.5	67.7	47.1	432B
◇MaxViT-B	640×640	51.6	71.3	56.1	44.1	68.5	47.7	543B
◇+$MULR_{896\rightarrow640}$	640×640	52.3	71.5	57.0	44.7	68.9	48.7	543B
◇MaxViT-T	896×896	52.1	71.9	56.8	44.6	69.1	48.4	475B
◇MaxViT-S	896×896	53.1	72.5	58.1	45.4	69.8	49.5	595B
◇MaxViT-B	896×896	53.4	72.9	58.1	45.7	70.3	50.0	856B

Table 6. **Comparison of two-stage object detection and instance segmentation on COCO2017.** All models are pretrained on ImageNet-1K.

mentation results comparing state-of-the-art ConvNets and vision Transformers. AP and AP^m denote box and mask average precision. We report the train and evaluation resolutions as well as their corresponding FLOPs as reference for model complexity. It may be seen that MaxViT suffers from noticeable performance drop if training resolution is lower. However, we observed that training with the MULLER resizer can further improve the performance across the board. Specifically, on MaxViT-B at 640×640, finetuning with MULLER gains 0.7 AP and 0.6 mask AP on the COCO validation set without any FLOPs overhead.

Image Quality Assessment. We base our experiment on the AVA dataset [28], which includes 250K images rated by amateur photographers. Each image in the dataset is associated with a histogram of ratings from an average of 200 raters. Image quality and aesthetic assessment is a task that is sensitive to downscaling [13], as downscaling may negatively impact visual quality attributes such as sharpness. We use the Earth Mover's Distance (EMD) as our training loss, similar to previous work of [44].

Our results are shown in Tab. 7. We report the Pearson linear correlation coefficient (PLCC) of the predicted and ground truth mean ratings as our evaluation metric. As can be seen, the proposed resizer improves the performance of MaxViT beyond the existing methods such as MUSIQ [13]. Note that in contrast to MUSIQ which uses multi-scale input augmentations, MULLER+MaxViT only requires a single low-resolution input from the resizer.

Model	Size	Params	PLCC↑
•NIMA [44]	224	56M	0.636
•+ $[45]_{512\rightarrow224}$	224	56M	0.680
•EffNet-B0 [46]	224	5.3M	0.642
•+ $[45]_{512\rightarrow224}$	224	5.3M	0.650
•AFDC [5]	224	44.5M	0.671
○ViT-S/32 [13]	384	22M	0.665
○ViT-B/32 [13]	384	88M	0.664
○MUSIQ [13]	224∼512	27M	0.720
◇MaxViT-T	224	31M	0.707
◇+$MULLER_{512\rightarrow224}$	224	31M	**0.729**

Table 7. **Image aesthetic assessment results on the AVA benchmark [28].** PLCC represents the Pearson's linear correlation coefficient.

nlayers	ksize	std	Top-1 Acc	FLOPs
2	5	1.0	85.58	24.23B
3	5	1.0	85.52	24.24B
4	5	1.0	85.50	24.25B
6	5	1.0	85.59	24.27B
2	3	1.0	85.48	24.23B
2	7	1.0	85.57	24.24B
2	5	1.5	85.56	24.24B
2	5	2.0	85.53	24.24B

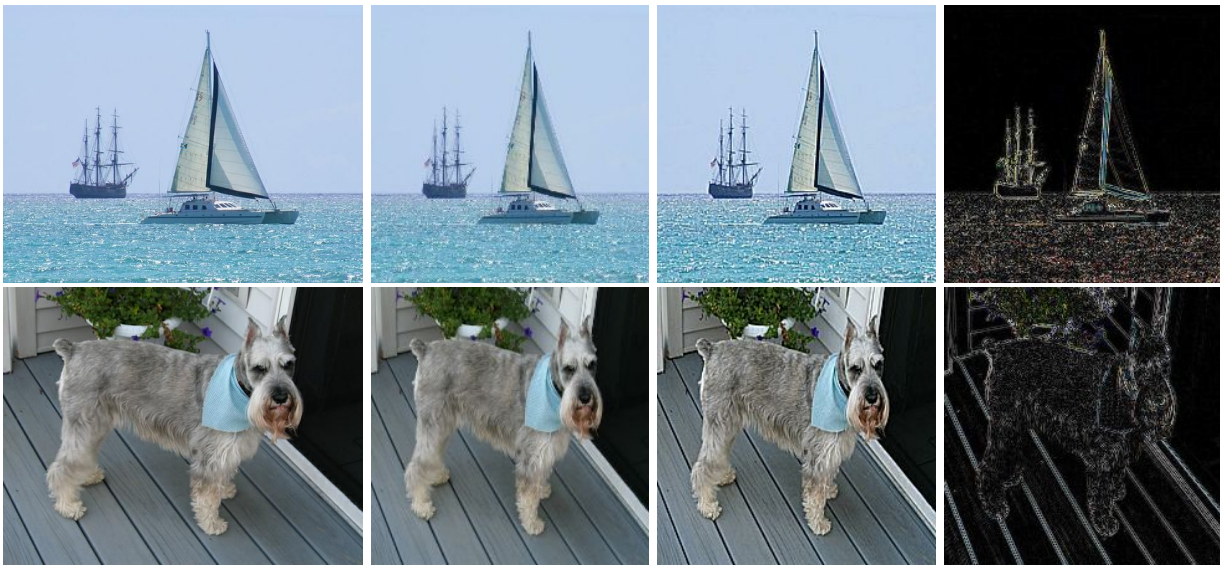
Table 8. **Hyperparameter sweep for MULLER.**

input size	output size	Top-1 Acc	FLOPs
384	224	85.45	24.22B
512	224	85.58	24.23B
768	224	85.56	24.26B
512	384	86.42	74.25B
768	384	86.44	74.28B
768	512	86.68	138.57B
1024	512	86.67	138.60B

Table 9. **Effects of input and output size for MULLER** using MaxViT-Base as test backkbone. Note that the ouput size is the image size seen by the backbone.

4.4. Ablation

Hyperparameters. There are three hyperparameters in the design of MULLER: $\{k, hsize, stddev\}$, which denote the number of layers, the kernel size of the Gaussian filters $\{\mathbf{W}_1, \mathbf{W}_2, ..., \mathbf{W}_k\}$, and the standard deviation of the Gaussian filters. To understand the effect of these hyperparameters, we conduct an ablation study. As shown in Tab. 8, we found that MULLER is quite insensitive to the selection



(a) Original (b) Default resizer (c) Learned resizer (d) Difference

Figure 4. Visualizations of the learned MULLER resizer for ResNet-50. Here the default resizer is an (anti-aliased) `AREA` resizer in Tensorflow. (d) shows the difference of the learned and the default resizers.

of parameters, and thus we recommend to use a simple set of parameters to save computation.

Effects of image size. It is known that image size can significantly affect the recognition performance. We evaluate the effect of varying input and output sizes of MULLER using MaxViT-B in Tab. 9. Note that the output size of MULLER is indeed the size seen by the backbone, so higher output size typically corresponds to improved accuracy. We observe that using higher input resolutions (*e.g.*, $3\times$) for MULLER does not yield any further performance gain beyond the baseline setting. Nonetheless, we find that adopting a reasonably large resolution (*e.g.* $1.6\sim2.5\times$) is necessary to achieve the expected performance. It is worth highlighting that this is perhaps impacted by the original resolution of images in the benchmark dataset.

4.5. Visualization

We visualize the behavior of the learned resizer in Fig. 4. As can be seen, the MULLER resizer is learned to boost details or textures of the images, while also enhancing the image contrast. These effects can preserve more visual information in the downscaled images over naive resizing, thus making the classification model learn better. As compared to the previous less-controlled resizer [45], MULLER achieves a better balance of human and machine perceptual qualities, due to the strong regularization imbued in its Laplacian-inspired design. We also point out that training MULLER with aliased inputs may produce relatively less sharper images in comparison to Fig. 4. We refer the reader to the supplementary material for visual examples.

5. Concluding Remarks

In this paper, we introduce MULLER, an extremely simple and light learned resizer, using multilayer Laplacian decomposition. The proposed resizer only contains 4 trainable parameters with negligible training and inference costs. This allows deploying the resizer as a thumbnail generator to produce optimally downscaled images for sending to remote inference servers, or alternatively as a server side resizer that reduces the inference cost without the necessity of changing the backbone architecture. We show that MULLER not only pushes forward the limit of the state-of-the-art vision Transformer MaxViT on ImageNet classification, but it also consistently improves across a range of widely used architectures, including EfficientNet, MobileNet, and ResNet. Additionally, we provide experiments to substantiate the efficacy of MULLER for various downstream tasks, such as object detection, segmentation, and image quality assessment. As compared to previous methods, MULLER enjoys remarkable generalization ability, owing to the strong regularization provided by its multilayer bandpass design. We believe that our work will inspire future research in this critically important direction: how to better preprocess images for vision tasks.

Limitations and future works. We note that if the higher-resolution inputs fail to boost the performance of an specific task, we cannot reasonably expect the learned resizer to provide a substantial performance boost either. Another potential future direction is to train a universal learned resizer that can be a drop-in replacement of the off-the-shelf resizers in existing machine learning frameworks, without necessitating the joint re-training of the backbones.

References

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

[2] Andy Brock, Soham De, Samuel L Smith, and Karen Simonyan. High-performance large-scale image recognition without normalization. In *International Conference on Machine Learning*, pages 1059–1071. PMLR, 2021. 6

[3] Li-Heng Chen, Christos G Bampis, Zhi Li, Lukáš Krasula, and Alan C Bovik. Estimating the resize parameter in end-to-end learned image compression. *arXiv preprint arXiv:2204.12022*, 2022. 2, 3

[4] Li-Heng Chen, Christos G Bampis, Zhi Li, Joel Sole, and Alan C Bovik. A progressive architecture for learned fractional downsampling. In *2021 Picture Coding Symposium (PCS)*, pages 1–5. IEEE, 2021. 2, 3

[5] Qiuyu Chen, Wei Zhang, Ning Zhou, Peng Lei, Yi Xu, Yu Zheng, and Jianping Fan. Adaptive fractional dilated convolution network for image aesthetics assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14114–14123, 2020. 7

[6] Wuyang Chen, Xianzhi Du, Fan Yang, Lucas Beyer, Xiaohua Zhai, Tsung-Yi Lin, Huizhong Chen, Jing Li, Xiaodan Song, Zhangyang Wang, and Denny Zhou. A simple single-scale vision transformer for object localization and instance segmentation. *CoRR*, abs/2112.09747, 2021. 7

[7] Zihang Dai, Hanxiao Liu, Quoc Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems*, 34, 2021. 5, 6

[8] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. *arXiv preprint arXiv:2107.00652*, 2021. 4, 5

[9] 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. *arXiv preprint arXiv:2010.11929*, 2020. 4, 5, 6

[10] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Task-driven super resolution: Object detection in low-resolution images. In *Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, December 8–12, 2021, Proceedings, Part V 28*, pages 387–395. Springer, 2021. 3

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 4, 5, 6, 7

[12] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 1

[13] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5148–5157, 2021. 7

[14] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *European conference on computer vision*, pages 491–507. Springer, 2020. 6

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 1, 4

[16] Boyi Li, Xiulian Peng, Zhangyang Wang, Jizheng Xu, and Dan Feng. End-to-end united video dehazing and detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 3

[17] Boyi Li, Wenqi Ren, Dengpan Fu, Dacheng Tao, Dan Feng, Wenjun Zeng, and Zhangyang Wang. Benchmarking single-image dehazing and beyond. *IEEE Transactions on Image Processing*, 28(1):492–505, 2018. 2

[18] Siyuan Li, Iago Breno Araujo, Wenqi Ren, Zhangyang Wang, Eric K Tokuda, Roberto Hirata Junior, Roberto Cesar-Junior, Jiawan Zhang, Xiaojie Guo, and Xiaochun Cao. Single image deraining: A comprehensive benchmark analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3838–3847, 2019. 2

[19] Yinxiao Li, Pengchong Jin, Feng Yang, Ce Liu, Ming-Hsuan Yang, and Peyman Milanfar. Comisr: Compression-informed video super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2543–2552, 2021. 2

[20] Yue Li, Dong Liu, Houqiang Li, Li Li, Zhu Li, and Feng Wu. Learning a convolutional neural network for image compact-resolution. *IEEE Transactions on Image Processing*, 28(3):1092–1107, 2018. 2

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 6

[22] Ding Liu, Bihan Wen, Xianming Liu, Zhangyang Wang, and Thomas S Huang. When image denoising meets high-level vision tasks: A deep learning approach. *arXiv preprint arXiv:1706.04284*, 2017. 3

[23] Feng Liu, Ronghang Zhu, Dan Zeng, Qijun Zhao, and Xiaoming Liu. Disentangling features in 3d face shapes for joint face reconstruction and recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5216–5225, 2018. 3

[24] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 1, 4, 5, 6, 7

[25] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*, 2022. 5, 6, 7

[26] Zhuang Liu, Tinghui Zhou, Hung-Ju Wang, Zhiqiang Shen, Bingyi Kang, Evan Shelhamer, and Trevor Darrell. Transferable recognition-aware image processing. *arXiv preprint arXiv:1910.09185*, 2019. 3

[27] Peyman Milanfar. A tour of modern image filtering: New insights and methods, both practical and theoretical. *IEEE signal processing magazine*, 30(1):106–128, 2012. 3

[28] Naila Murray, Luca Marchesotti, and Florent Perronnin. Ava: A large-scale database for aesthetic visual analysis. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2408–2415. IEEE, 2012. 7

[29] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On buggy resizing libraries and surprising subtleties in fid calculation. *arXiv preprint arXiv:2104.11222*, 2021. 2, 6

[30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 1

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. 6

[32] Rachid Riad, Olivier Teboul, David Grangier, and Neil Zeghidour. Learning strides in convolutional neural networks. *arXiv preprint arXiv:2202.01653*, 2022. 2

[33] Avinab Saha, Sandeep Mishra, and Alan C Bovik. Re-iqa: Unsupervised learning for image quality assessment in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5846–5855, 2023. 1

[34] Mehdi SM Sajjadi, Bernhard Schölkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. *arXiv preprint arXiv:1612.07919*, 2016. 2

[35] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 1, 4, 5, 6

[36] Vivek Sharma, Ali Diba, Davy Neven, Michael S Brown, Luc Van Gool, and Rainer Stiefelhagen. Classification-driven dynamic image enhancement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4033–4041, 2018. 3

[37] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016. 2

[38] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017. 4

[39] Satoshi Suzuki, Motohiro Takagi, Kazuya Hayase, Takayuki Onishi, and Atsushi Shimizu. Image pre-transformation for recognition-aware image compression. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 2686–2690. IEEE, 2019. 3

[40] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 1

[41] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 1

[42] Hossein Talebi, Damien Kelly, Xiyang Luo, Ignacio Garcia Dorado, Feng Yang, Peyman Milanfar, and Michael Elad. Better compression with deep pre-editing. *IEEE Transactions on Image Processing*, 30:6673–6685, 2021. 3

[43] Hossein Talebi and Peyman Milanfar. Fast multilayer laplacian enhancement. *IEEE Transactions on Computational Imaging*, 2(4):496–509, 2016. 4

[44] Hossein Talebi and Peyman Milanfar. Nima: Neural image assessment. *IEEE transactions on image processing*, 27(8):3998–4011, 2018. 7

[45] Hossein Talebi and Peyman Milanfar. Learning to resize images for computer vision tasks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 497–506, 2021. 2, 3, 5, 6, 7, 8

[46] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 1, 4, 5, 6, 7

[47] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International Conference on Machine Learning*, pages 10096–10106. PMLR, 2021. 5, 6

[48] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8174–8182, 2018. 2

[49] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxim: Multi-axis mlp for image processing. *arXiv preprint arXiv:2201.02973*, 2022. 2

[50] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pages 459–479. Springer, 2022. 1, 2, 4

[51] Cristina Vasconcelos, Hugo Larochelle, Vincent Dumoulin, Rob Romijnders, Nicolas Le Roux, and Ross Goroshin. Impact of aliasing on generalization in deep convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10529–10538, 2021. 6

[52] Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling

local self-attention for parameter efficient visual backbones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12894–12904, 2021. 6

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[54] Zhou Wang. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 2

[55] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22–31, 2021. 6

[56] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 7

[57] Runsheng Xu, Jinlong Li, Xiaoyu Dong, Hongkai Yu, and Jiaqi Ma. Bridging the domain gap for multi-agent perception. *arXiv preprint arXiv:2210.08451*, 2022. 2

[58] Runsheng Xu, Hao Xiang, Zhengzhong Tu, Xin Xia, Ming-Hsuan Yang, and Jiaqi Ma. V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*, pages 107–124. Springer, 2022. 1

[59] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017. 2

[60] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 649–666. Springer, 2016. 2

[61] Chen Zhao and Bernard Ghanem. Thumbnet: One thumbnail image contains all you need for recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1506–1514, 2020. 2