One Size Fits All: Hypernetwork for Tunable Image Restoration

Shai Aharon$^{1}[0000−0003−0877−9699]$ and Dr. Gil Ben-Artzi$^{1}[0000−0001−7449−5163]$

Ariel University, Ariel, Israel ifryed@gmail.com, gilba@ariel.ac.il

Abstract. We introduce a novel approach for tunable image restoration that achieves the accuracy of multiple models, each optimized for a different level of degradation, with exactly the same number of parameters as a single model. Our model can be optimized to restore as many degradation levels as required with a constant number of parameters and for various image restoration tasks. Experiments on real-world datasets show that our approach achieves state-of-the-art results in denoising, DeJPEG and super-resolution with respect to existing tunable models, allowing smoother and more accurate fitting over a wider range of degradation levels.

Keywords: Deep Learning, Hyper networks, Image restoration, Tunable, Super resolution, Denoise, DeJPEG

1 Introduction

Deep learning has demonstrated state of the art results in many image restoration tasks, such as denoising [31,32] and super-resolution [18,33]. The common approach is to train the model in a supervised manner, optimizing the model for only a single degradation level. In practice, the exact degradation level of the corrupted image is not known a-priori. Blind image restoration [6] addresses this issue by learning to predict the specific degradation kernel for the corrupted image to restore its clean version. Recently, tunable image restoration has gained popularity as an alternative. In this approach, the user can tune the restoration effect to generate a variety of output images and select one according to his own preferences. They enable image- and user-specific adaptation for all degradation levels with a single model without the need for retraining or deploying multiple models.

Various approaches have been presented for such models [11,29,27,28,10]. State-of-the-art approaches train the network to fit two degradation levels instead of only one, an initial and final one that span the required range of degradation levels. At inference time, an input parameter is used to adjust the model’s weights to an unseen degradation level. In practice, however, accuracy declines as one moves from the trained to the untrained level. As a result, several models need to be trained in order to achieve accuracy across the entire range.

We seek a model that is highly accurate, equivalent to multiple independent models, without any additional increase in the size of the base model. The mere
Fig. 1: Our model achieves higher restoration accuracy with the size of only a single base model without any increase in the number of parameters, compared to AdaFM [11].

existence of such a single model that can match the accuracy of multiple dedicated models down to a negligible difference may seem unexpected. We argue in this study that it is possible to simultaneously optimize the distinct weights associated with each kernel for various degradation levels and learn a single instance of basis vectors that span the weights. The result is a single, highly efficient and accurate model as all degradation levels can be accounted for through the use of linear interpolation coefficients and a single instance of basis vectors.

To this end, we introduce a hypernetwork that learns to generate the filter weights of an image restoration network conditionally based on the required restoration level, given as an input parameter. To ensure an efficient representation at inference time, we constrain the kernels corresponding to the different levels to be identical up to a scaling factor and learn only the weights’ basis vectors. Our model is trained to generate multiple restoration networks simultaneously; each is fitted to a different degradation level. In order to achieve the desired accuracy and continuity over any predefined range, it can be fitted and tuned to as many degradation levels as required. During inference, it can be easily used to efficiently generate the required filter weights on-the-fly. Fig. 1 shows the result of our approach and AdaFM [11] for super-resolution. Using our approach we can tune the model to achieve higher accuracy with only a small amount of parameters.

Contribution. We introduce a hypernetwork model for tunable image restoration tasks, which can accommodate a wide range of degradation levels, and unlike previous approaches, does not sacrifice size or accuracy. It is practically as accurate as multiple dedicated models, regardless of the range of degradation levels supported. It has the same footprint as a single restoration network model and requires no additional parameters. It outperforms existing state-of-the-art tunable restoration models in both accuracy and size, and can be easily deployed to generate at runtime state-of-the-art image restoration networks.
2 Related Work

Modulating Networks. Recently, there has been a growing interest in constructing networks which can be continuously tuned at inference time. These can broadly be categorized into two categories, models which allow tuning different objectives at inference time and models which allows different restoration levels of the same objective, where our approach falls into the latter category. The typical approach is to train two related networks on different objectives and apply interpolation between their weights. The networks can be either the same or with additional tuning blocks. Dynamic-Net [27] adds specialized blocks directly after the convolution layers, which are optimized during the training to the additional objective. CFSNet [28] used branches, each is based on a different objective. AdaFM [11] added modulation filters after each convolution layer. DNI [29] train the same network architecture on the different objective and interpolates all the parameters. Son [17] extended the approach of [11] with an FTN module allowing better non-linear interpolation.

These methods are mainly focused on advanced modulation modules and techniques to improve the model’s accuracy. Our goal is to develop a model that can be scaled accurately to a variety of degradation levels in the most efficient manner possible.

Weight-generating network. Learning to learn, or meta learner, uses meta networks to generate weights for the main network for various tasks [25,19,15]. Hypernetwork, introduced in [9], uses a small network with a reduced number of parameters to generate the weights for a larger target network. It often uses weight sharing across layers, while reaching accurate results. [4] applied hypernetwork to Neural Architecture Search (NAS) [22,7] aimed to create or improve the search of the whole network architecture. It was also used in Bayesian context [16], and in Stochastic maximum likelihood optimization [23]. [8] presented an image restoration hypernetwork with a single main network. In our approach, we employ a hypernetwork to generate the weights of kernels in multiple target networks, such that they are linearly depended.

3 Our Approach

Our key contribution is the deployment of a hypernetwork to jointly fit image restoration networks to multiple degradation levels while learning the basis vectors for the weights of their convolutional kernels so that they differ only by a single scalar. The expressiveness of such an architecture enables it to achieve the best accuracy across a broad range of degradation levels and restoration tasks, with a network size of only a single network at inference time.

3.1 Network Architecture

Overview. Our proposed model is illustrated in Fig. It consists of a hypernetwork $h$ and main restoration networks $n_i$. The weights of our hypernetwork
Fig. 2: Our hypernetwork consists of $l$ meta blocks, one for each convolutional layer. Each meta block is a Fully Connected (FC) layer which outputs the kernels' weights for the corresponding kernel in the main network.

$\theta^h$ are fixed during inference time and learned during the training process, while $\theta^{n_i}$, the weights of the restoration network $n_i$, are dynamically generated at inference time. The input to hypernetwork $h$ is an encoding scalar $\bar{c}_i \in \mathbb{R}$ corresponds to a degradation level $c_i$, and the outputs are the kernels’ weights for the main restoration networks $n_i$. The encoding scalar is a normalized degradation level. We set the normalization factor to $\frac{1}{255}$, in all the experiments across all datasets and tasks, without fine-tuning it for each task individually.

The main network is a standard image restoration network [5]. It consists of a downsampling layer using convolution with a stride of 2, 16 residual [12] blocks and upsampling layers using pixelshuffle [26] and a skip-connection over the residual blocks. During training, each main network $n_i$ is optimized to restore a degraded image with corresponding degradation level $c_i$. The input is a degraded image $I^{c_i} \in \mathbb{R}^{3 \times H \times W}$ with a degradation level $c_i$ and the output is the restored image with the same dimensions. The weights of each main network are the weights of the residual block’s kernels generated by the hypernetwork (Fig. 3, green background) and the weights of the head and tail of the network which are shared among all the main networks (Fig. 3, yellow background).

The hypernetwork $h$ consists of $l$ meta blocks, where $l$ is the number of kernels in the main network’s residual blocks (Fig. 2). Meta block $m^j$ is a fully connected layer constructed out of weights and bias $w^j, b^j \in \mathbb{R}^{(C_{out} \times C_{in} \times K \times K) \times 1}$ to adaptively generate the weights of the $j^{th}$ kernel of main network $n_i$ by:

$$k^j_i = c_i w^j + b^j,$$

where $C_{in}$ and $C_{out}$ are the number of input and output channels, respectively, and $K \times K$ is the kernel’s size. $k^j_i$ is the kernel in the flattened form. Unlike hypernetworks [9], our method assigns each meta block to generate weights for a specific main network layer with one common input scalar. Due to the bias term, the output convolutional kernels are not identical throughout the various main networks up to the input scalar ($c_i$).
Training. Our model learns \( l \) shared weights \( \{(w^j, b^j)\}_{m=1}^l \) for \( k \) degradation levels by jointly optimizing the \( l \) meta blocks (basis vectors) and the \( k \) main networks. The number of main networks \( (k) \) is fixed during the training process. Each image in the training set \( D = \{I_1, I_2, \ldots, I_n\} \) is degraded with \( k \) degradation levels \( \{c_1, c_2, \ldots, c_k\} \) and fed into the corresponding main network \( \{n_1, n_2, \ldots, n_k\} \). Each degradation level \( c_i \) is mapped to a fully parameterized main network \( n_i \) through the encoding scalar and meta blocks. Our objective is to optimize the overall restoration accuracy under the different degradation levels. Therefore, no degradation level is privileged and our total loss is the unweighted sum of individual \( L_1 \) losses, represented as:

\[
L_{total} = \frac{1}{k} \sum_{i=1}^k \sum_{j=1}^n L_1(\theta^{n_i}; I^{c_i}_j, I_j).
\] (2)

Since the aforementioned weight generation operations are completely differentiable, the parameters in our hypernetwork \( h \) are optimized simultaneously following the chain rule.

Inference. Given a degraded image and an input degradation level \( c_i \), we employ the learned weights of the hypernetwork \( \theta^h \) to generate the weights of a restoration network \( \theta^{n_i} \). Each meta block generates the weights of according to Eq. 1. A simple user interface enables the user to interact with the system.
in real time, selecting the input value and, as a result, the desired restoration outcome. The restoration network generation is efficient involving multiplication of the same single scalar for all the residual block’s kernels of the main network. We demonstrate in the next section that the various restoration networks formed by the multiplication of the different scalars can accurately restore the degraded images for a variety of degradation levels.

4 Experiments

4.1 Model Training

We used the DIV2K dataset to train the models for denoising, Deblocking JPEG (DeJPEG) and super-resolution. It includes 1000 2K resolution RGB images, with 800 images used for training, 100 for validation and 100 for testing. The training data was augmented by standard practice, with both horizontal flipping and rotations. The mini-batch size is set to 16. We used randomly cropped 96 × 96 patches from each image as our training data. The $L_1$ loss was used throughout all the experiments. We heuristically set the weight of each main network’s loss uniformly. We used an initial learning rate set to $1 \times 10^{-4}$, decaying by a factor of 10 after $5 \times 10^5$ iterations. We trained our model for $1 \times 10^6$ iterations in total. The Adam optimizer was deployed with $\beta_1 = 0.9, \beta_2 = 0.999$. We implemented our approach using PyTorch library. The model was trained on a NVIDIA RTX 2080Ti GPU for approximately 12 hours.

Our model includes a single hypernetwork and multiple main networks. Each main network is trained to optimally restore a different degradation level. The hypernetwork generates weights simultaneously for all the main networks, each is conditionally based on the corresponding degradation level the network is optimized for. The input degradation level is normalized by 255 for all tasks, showing the robustness of our model. For denoising, we trained a model with four main networks to restore noise levels $\sigma = 15, 35, 50, 75$. The input for each main network is a ground truth patch corrupted by the corresponding Gaussian noise level. The output of each main network is evaluated with respect to the same ground truth input patch. For DeJPEG, we train our model with five main networks to restore compression qualities $q = 10, 25, 45, 65, 80$ using OpenCV JPEG encoder, where 80 has minor artifacts, and 10 has severe high frequency loss resulting in noticeable artifacts. Similar to [11,29], we use grayscale images for training and testing. For super-resolution, we downsampled images by various factors and upsampled them to the original size using OpenCV bicubic interpolation (Pre-Upscaling SR [30]). We train our model with four main networks to restore upscaling factors of $\times 2, \times 3, \times 4, \times 5$.

4.2 Restoration Quality

We evaluate our approach for the following tasks - denoising, DeJPEG and super-resolution (Fig. 1). For denoising we use the CBSD68 dataset [20], for DeJPEG
Table 1: Results for image denoising. Top table is PSNR and bottom table is SSIM. Baselines are individual models trained to restore a specific noise level. For most noise levels in the range, our approach achieves comparable accuracy to individual models, allowing accurate and fast restoration at inference time of a wide range of degradation levels with number of parameters of only a single model.

|       |  5  | 15  | 25  | 35  | 45  | 55  | 65  | 75  | 85  |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Baseline | 40.48 | 34.08 | 31.42 | 29.80 | 28.64 | 27.78 | 27.06 | 26.46 | 25.95 |
| DNI | 23.64 | 34.07 | 25.44 | 22.26 | 21.41 | 22.08 | 24.24 | 26.46 | 25.01 |
| AdaFM | 38.80 | **34.08** | 31.19 | 29.30 | 28.42 | 27.40 | 26.89 | 26.34 | 25.61 |
| DeCoupling | 34.40 | 30.87 | 27.65 | 25.46 | 23.89 | 22.74 | 21.87 | 21.22 | 20.72 |
| CFSNet | 29.43 | 29.27 | 28.91 | 28.38 | 27.78 | 27.16 | 26.58 | 26.03 | 25.55 |
| Ours | **39.96** | 34.02 | **31.41** | **29.80** | **28.64** | **27.78** | **27.06** | **26.46** | **25.95** |
| Distance (Ours) | 0.52 | 0.06 | 0.01 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

|       |  5  | 15  | 25  | 35  | 45  | 55  | 65  | 75  | 85  |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Baseline | 0.98 | 0.93 | 0.89 | 0.85 | 0.81 | 0.78 | 0.76 | 0.74 | 0.71 |
| DNI | 0.95 | **0.95** | 0.87 | 0.81 | 0.77 | 0.76 | 0.75 | **0.73** | **0.71** |
| AdaFM | 0.97 | **0.95** | 0.88 | 0.84 | 0.80 | 0.77 | 0.75 | **0.73** | 0.70 |
| DeCoupling | 0.97 | 0.89 | 0.78 | 0.69 | 0.61 | 0.55 | 0.51 | 0.47 | 0.45 |
| CFSNet | 0.84 | 0.83 | 0.82 | 0.81 | 0.78 | 0.76 | 0.73 | 0.71 | 0.68 |
| Ours | **0.98** | **0.95** | **0.89** | **0.85** | **0.81** | **0.78** | **0.76** | **0.73** | **0.71** |

the LIVE1 [24] and for super-resolution Set5 [2]. In all tasks the baselines are independently trained restoration models specifically optimized to restore a single degradation level. To compare our accuracy with respect to a single tunable model (same as ours), we also included state-of-the-art tunable models, AdaFM [11], DNI [29] and CFSNet [28]. For denoising, they were trained on $\sigma = 15, 75$. For DeJPEG, they were trained on $q = 10, 80$. For super-resolution, they were trained on $\times 2, \times 4$. In addition, we also compare with respect to Decouple learning [8], trained on the same objectives as ours. For each task we report standard metrics - PSNR and SSIM [13]. In all tables bold represents the best results. For super-resolution, as commonly done, we train our model based on RGB images and evaluate the PSNR based on the y-channel. During evaluation, our trained hypernetwork generates the weights of the main network corresponding to the input degradation level, which is then used to restore the images.

For denoising, we evaluate our model with respect to all noise levels from 5 to 90 with intervals of 5. Both AdaFM and DNI share the same network structure as our standard image restoration network, with 16 residual blocks where Decouple learning consists of 20 blocks. Table 1 shows our main results for the denoising task, with restoration accuracy measured by PSNR and SSIM. It can be seen that our technique outperforms all other methods in almost all
Table 2: Results for super resolution task.

| Method   | PSNR (dB) | SSIM   |
|----------|-----------|--------|
|          | 2 3 4 5 6 |        |
| Baseline | 36.95 29.86 29.54 25.67 25.06 | 0.94 0.84 0.84 0.74 0.71 |
| DNI      | 24.14 23.08 23.06 21.63 21.54 | 0.74 0.79 0.76 0.68 0.65 |
| AdaFM    | 36.95 28.85 27.87 25.15 24.38 | 0.94 0.83 0.81 0.57 0.03 |
| DeCoupling | 29.97 26.35 25.90 23.45 22.76 | 0.92 0.83 0.81 0.72 0.68 |
| CFSNet   | 30.47 26.29 25.40 22.90 22.30 | 0.92 0.82 0.78 0.69 0.65 |
| Ours     | **36.69** 29.77 29.46 **25.63** 24.92 | **0.94** **0.84** **0.83** **0.74** **0.71** |

Table 3: Results for DeJPEG artifacts removal task.

| Method   | PSNR (dB) | SSIM   |
|----------|-----------|--------|
|          | 10 30 50 70 80 |        |
| Baseline | 28.82 22.57 34.40 36.40 38.14 | 0.82 0.91 0.94 0.96 0.97 |
| DNI      | 25.76 21.58 15.84 23.39 **38.16** | 0.82 0.86 0.87 **0.94** **0.97** |
| AdaFM    | 27.15 31.28 33.59 36.22 38.14 | 0.77 0.86 0.87 0.89 0.90 |
| DeCoupling | 28.25 31.68 33.28 35.04 36.44 | 0.81 **0.91** 0.93 0.95 0.96 |
| CFSNet   | 27.63 32.12 33.72 34.88 35.48 | 0.80 0.92 0.94 0.95 0.96 |
| Ours     | **28.81** **32.56** **34.39** **36.38** **38.09** | **0.82** **0.91** **0.94** **0.96** **0.97** |

noise levels tested, and that it achieves restoration accuracy equivalent to that of dedicated models, identically in most of the range. The average PSNR distance from the optimal accuracy is 0.009. Interestingly, even when applying our model to the high noise level range of 75 – 90, our model exhibits high accuracy with an average PSNR distance of 0.006. For the lower range of noise levels, our model obtained an average PSNR distance of 0.325. To obtain a lower PSNR distance, it is possible to extend the training of our model for additional degradation levels, as discussed in the next section.

For super-resolution, we evaluate our model on five upscaling factors, ×2, ×3, ×4, ×5, ×6. As before, baselines consist of multiple independent models that are each trained to fit a particular degradation level. Table 2 shows our results. Similar to denoising, our method outperforms all other methods, with a significant PSNR difference from the second-best method.

For DeJPEG, we evaluate our model on eight different compression levels. Table 3 shows our results with respect to optimal accuracy obtained by training independent models to restore each compression level, AdaFM, DNI, CFSNet and Decouple learning. Our approach achieved comparable PSNR accuracy to the optimal ones at all compression levels. It also achieved the highest SSIM for all compression levels. Fig. 9 shows how our methods performed on various compressed images.
4.3 Size vs. Accuracy

Figure 5 presents the accuracy achieved by our model at inference time with respect to the number of residual blocks (and size) of the trained restoration network. With only two residual blocks, our model achieves 98.6% of the optimal PSNR. Using four residual blocks, our model obtained slightly better accuracy than the current state-of-the-art model \[11\], using only \(\frac{1}{125}\) of the parameters. This implies that optimizing the weights of multiple networks simultaneously can improve model’s generalization. Using all sixteen residual blocks, our approach can achieve state-of-the-art accuracy.

4.4 Blind Image Restoration

A tunable image restoration model is designed to provide the user with the ability to adjust the restoration output at runtime. In order to provide a restoration
output corresponding to the input degradation level when the level is unknown, we trained a simple CNN (five convolutional layers and three fully connected layers) to estimate the degradation level of a noisy image (see Supplementary for further details). Based on our trained network, we can estimate the input degradation level and set the encoding scalar $c_i$ accordingly. Fig. 6 and 7 demonstrate the accuracy of the noise estimation network. Fig. 6 shows the error of the estimation as a percentage of the ground truth noise level. On average, the degradation level estimation network achieves an accuracy of 98.23%. Fig. 7 obtained by setting the input degradation level for each degraded image according to the estimated degradation level, shows the effect of the estimated degradation level on the final restoration accuracy of our model. Overall, the estimation of noise level results in accurate restoration and can be advantageous in cases where the actual degradation level is unknown.

### 4.5 Range of degradation levels

Our approach provides the ability to take into account a wide range of degradation levels without increasing the size or compromising its accuracy at inference.
Table 4: The architecture of our model with and without bias and batch normalization. We use the 4 main networks, $\sigma = 15, 35, 50, 75$, as the baseline.

|        | Baseline | With BN | With Bias | BN and Bias |
|--------|----------|---------|-----------|-------------|
| $\sigma$ |          |         |           |             |
| 5      | 40.48    | 34.50   | 39.96     | 35.46       |
| 15     | 34.08    | 33.97   | 34.02     | 33.98       |
| 25     | 31.42    | 31.35   | 31.41     | 31.30       |
| 35     | 29.80    | 29.76   | 29.80     | 29.77       |
| 45     | 28.64    | 28.61   | 28.64     | 28.61       |
| 55     | 27.78    | 27.73   | 27.78     | 27.73       |
| 65     | 27.06    | 27.01   | 27.06     | 27.00       |
| 75     | 26.46    | 26.43   | 26.46     | 26.44       |
| 85     | 25.95    | 22.38   | 25.95     | 23.20       |

We trained our model for two ranges of degradation levels: $15 - 75$ and $15 - 100$, both based on four main networks ([15, 40, 70, 100], [15, 35, 50, 75]). In spite of the increased range, both models achieve 99% accuracy compared to the optimal accuracy. Both approaches require the same number of parameters during inference - only a single model.

4.6 Tuning Accuracy

We test the ability of the hypernetwork to generate the optimal weights for a given input degradation level. For each image in the test set, we degraded the image with a specific degradation level, e.g. $\sigma = 15$ for the denoising task. For the degraded image we measure the best input parameter that yields the network with highest restoration accuracy in terms of PSNR. Fig. 8 presents both the ground truth degradation level and the best input degradation level with respect to different levels for the denoising (top) and DeJPEG (bottom). As can be seen, our approach achieves high levels of restoration quality by generating weights according to the input degradation level. Similar results obtained for both DeJPEG and super-resolution. Figure 9 shows the result of tuning our model for DeJPEG task. The restoration effect is according to the input degradation level, as required.

4.7 Ablation Study

**Batch Normalization and Bias.** We explore the optimal architecture of our default main restoration network with respect to bias and batch normalization [14] (BN). For batch normalization testing, we train a network with BNs after all convolutional layers and adjust the statistics in BN layers during testing. The results are shown in Table 4. It can be seen that the bias is an essential ingredient in our approach. Using batch normalization without bias does not result in a successful restoration. Batch normalization used with bias produces comparable results to bias-only in the range 15-75. We deploy the bias-only configuration as our default one due to its simplicity.

**Number of Main Networks.** We explore the number of main networks in the model and its effect on the performance. We tested various configurations...
Fig. 8: Tuning accuracy. The dark green bars represent the ground truth degradation levels (the numbers above) while the light green bars represent the mean value of the best input parameter that achieved the highest restoration accuracy. Top row is denoising and bottom row is DeJPEG. It can be seen that our tuning is accurate.

for our model, from one to five different main networks. Note that the number of main networks does not change the final size of our model at inference, only at training.

For the denoising task, for the model with two networks we use $\sigma = [5, 75]$. For the model with three main networks we used $\sigma = [15, 45, 85]$. For the model with four main networks we used $\sigma = [15, 35, 50, 75]$ and for the model with five main networks we used $\sigma = [5, 25, 45, 65, 85]$. Table 6 presents the PNSR obtained by each model. For the range of $15 - 85$, the model trained with four main networks is comparable to the model trained with five main networks. For noise level 5, the model trained with five main networks obtained higher accuracy. We further explore the performance of our approach with four and five main networks with respect to tunable restoration network with higher number of layers. We report results for CResMD [10] which includes almost twice as many layers (32) as our network. Although CResMD learns multiple types of degradation, it has been shown to outperform state-of-the-art methods for tunable denoising. Table 5 shows the results of our approach and CResMD for the denoising task. Overall, CResMD is better than both AdaFM and DNI and our approach outperforms all three. It can be seen that the four and five networks’ models perform similarly apart from the lowest noise level. This suggests that
For obtaining higher accuracy along a wider range, one might need to train a model with sufficient number of main networks.

For super-resolution we tested $[\times 2, \times 4]$, $[\times 2, \times 3, \times 4]$ and $[\times 2, \times 3, \times 4, \times 5]$. Figure 10 shows the results for the super-resolution task. Overall, the results indicate that by increasing the number of main networks to four, the accuracy is significantly improved. Due to the fact that at inference time we deploy a small-size single model regardless of the number of main networks trained, adding more main networks is beneficial.

DeJPEG was trained with three different models ($q = [10, 40, 80]$, $q = [10, 30, 50, 80]$ and $q = [10, 25, 45, 65, 80]$ ) and we obtained similar results as to previous tasks. The accuracy of restoration increases with the number of main networks.

Table 5: Our 16-layers approach trained to fit 5 noise level vs. 32-layers adaptive CResMD

| PSNR (dB) | 5 | 15 | 25 | 35 | 45 | 55 | 65 | 75 | 85 |
|-----------|---|----|----|----|----|----|----|----|----|
| Baseline  | 40.48 | 34.08 | 31.42 | 29.80 | 28.64 | 27.78 | 27.06 | 26.46 | 25.95 |
| CResMD    | 40.15 | 33.93 | 31.30 | 29.67 | 28.52 | 27.63 | 26.89 | 26.23 | 25.60 |
| Ours (5)  | 40.31 | 34.00 | 31.39 | 29.78 | 28.64 | 27.77 | 27.06 | 26.46 | 25.95 |
Fig. 10: The effect of the number of main networks for super-resolution. The x-axis is the number of objectives our model trained to adapt. The y-axis is the average error in percentages relative to multiple optimal models trained to restore the correspondent upscaling. Our model can achieve the optimal accuracy by increasing the number of main networks to four.

Table 6: The results for our image denoising models, each is trained with a different number of main networks.

| PSNR (dB) | 5    | 15   | 25   | 35   | 45   | 55   | 65   | 75   | 85   |
|----------|------|------|------|------|------|------|------|------|------|
| Baseline | 40.48| 34.08| 31.42| 29.80| 28.64| 27.78| 27.06| 26.46| 25.95|
| 2 main networks | 36.46| 34.02| 29.94| 27.66| 26.95| 26.89| 26.82| 26.46| 25.12|
| 3 main networks | 39.55| 34.02| 31.39| 29.78| 28.64| 27.77| 27.06| 26.46| 25.94|
| 4 main networks | 39.96| 34.02| 31.41| 29.80| 28.64| 27.78| 27.06| 26.46| 25.95|
| 5 main networks | 40.31| 34.00| 31.39| 29.78| 28.64| 27.77| 27.06| 26.46| 25.95|

5 Conclusion

The task of real-world image restoration is challenging because the exact input degradation level and the user’s preferences for the restoration effect are unknown. Current approaches compromise either accuracy, the range of the degradation levels they support, or the size of their solution. The combination of all three is imperative for a fully functional real-world solution, but such a method is currently unavailable. We introduce a hypernetwork for tunable image restoration that is accurate, maintains a small footprint during inference and can support a wide range of degradation levels. It is capable of dynamically generating the weights of a restoration network to best match a given level of degradation. We demonstrated that our approach achieved state-of-the-art results when compared to existing methods for both denoising, DeJPEG, and super-resolution tasks. The proposed design may be beneficial in a variety of applications, potentially enhancing existing approaches.
References

1. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (July 2017)
2. Bevilacqua, M., Roumy, A., Guillemot, C., line Alberi Morel, M.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: Proceedings of the British Machine Vision Conference. pp. 135.1–135.10. BMVA Press (2012). https://doi.org/http://dx.doi.org/10.5244/C.26.135
3. Bräsig, G.: The OpenCV Library. Dr. Dobb’s Journal of Software Tools (2000)
4. Brock, A., Lim, T., Ritchie, J.M., Weston, N.: Smash: One-shot model architecture search through hypernetworks (2017)
5. Dong, C., Deng, Y., Loy, C.C., Tang, X.: Compression artifacts reduction by a deep convolutional network (2015)
6. Elron, N., Yuval, S.S., Rudoy, D., Levy, N.: Blind image restoration without prior knowledge (2020)
7. Elsken, T., Metzen, J.H., Hutter, F.: Neural architecture search: A survey. The Journal of Machine Learning Research 20(1), 1997–2017 (2019)
8. Fan, Q., Chen, D., Yuan, L., Hua, G., Yu, N., Chen, B.: Decouple learning for parameterized image operators. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 442–458 (2018)
9. Ha, D., Dai, A., Le, Q.V.: Hypernetworks (2016)
10. He, J., Dong, C., Qiao, Y.: Interactive multi-dimension modulation with dynamic controllable residual learning for image restoration. arXiv preprint arXiv:1912.05293 (2019)
11. He, J., Dong, C., Qiao, Y.: Modulating image restoration with continual levels via adaptive feature modification layers. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
13. Horé, A., Ziou, D.: Image quality metrics: Psnr vs. ssim. In: 2010 20th International Conference on Pattern Recognition. pp. 2366–2369 (2010). https://doi.org/10.1109/ICPR.2010.579
14. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift (2015)
15. Kar, A., Prakash, A., Liu, M.Y., Cameracci, E., Yuan, J., Rusiniak, M., Acuna, D., Torralba, A., Fidler, S.: Meta-sim: Learning to generate synthetic datasets. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4551–4560 (2019)
16. Krueger, D., Huang, C.W., Islam, R., Turner, R., Lacoste, A., Courville, A.: Bayesian hypernetworks. arXiv preprint arXiv:1710.04759 (2017)
17. Lee, H., Kim, T., Son, H., Baek, S., Cheon, M., Lee, S.: Smoother network tuning and interpolation for continuous-level image processing. arXiv preprint arXiv:2010.02270 (2020)
18. Lim, B., Son, S., Kim, H., Nah, S., Mu Lee, K.: Enhanced deep residual networks for single image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 136–144 (2017)
19. Liu, B., Rao, Y., Lu, J., Zhou, J., Hsieh, C.J.: Metadistiller: Network self-boosting via meta-learned top-down distillation. In: European Conference on Computer Vision. pp. 694–709. Springer (2020)
16 A. G.

20. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001. vol. 2, pp. 416–423. IEEE (2001)

21. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

22. Ren, P., Xiao, Y., Chang, X., Huang, P.Y., Li, Z., Chen, X., Wang, X.: A comprehensive survey of neural architecture search: Challenges and solutions. arXiv preprint arXiv:2006.02903 (2020)

23. Sheikh, A.S., Rasul, K., Merentitis, A., Bergmann, U.: Stochastic maximum likelihood optimization via hypernetworks. arXiv preprint arXiv:1712.01141 (2017)

24. Sheikh, H.R., Sabir, M.F., Bovik, A.C.: A statistical evaluation of recent full reference image quality assessment algorithms. IEEE Transactions on Image Processing 15(11), 3440–3451 (2006). https://doi.org/10.1109/TIP.2006.881959

25. Shen, F., Yan, S., Zeng, G.: Neural style transfer via meta networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8061–8069 (2018)

26. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: 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. pp. 1874–1883 (2016)

27. Shoshan, A., Mechrez, R., Zelnik-Manor, L.: Dynamic-net: Tuning the objective without re-training for synthesis tasks. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)

28. Wang, W., Guo, R., Tian, Y., Yang, W.: Cfsnet: Toward a controllable feature space for image restoration (2019)

29. Wang, X., Yu, K., Dong, C., Tang, X., Loy, C.C.: Deep network interpolation for continuous imagery effect transition (2018)

30. Wang, Z., Chen, J., Hoi, S.C.: Deep learning for image super-resolution: A survey. IEEE transactions on pattern analysis and machine intelligence 43(10), 3365–3387 (2020)

31. Xie, J., Xu, L., Chen, E.: Image denoising and inpainting with deep neural networks. Advances in neural information processing systems 25, 341–349 (2012)

32. Yu, Z., Cui, Y., Yu, J., Wang, M., Tao, D., Tian, Q.: Deep multimodal neural architecture search. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 3743–3752 (2020)

33. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2472–2481 (2018)

34. Zhao, H., Gallo, O., Frisio, I., Kautz, J.: Loss functions for image restoration with neural networks. IEEE Transactions on Computational Imaging 3(1), 47–57 (2017). https://doi.org/10.1109/TCI.2016.2644865