Searching for Controllable Image Restoration Networks

Heewon Kim Sungyong Baik Myungsub Choi* Janghoon Choi† Kyoung Mu Lee
ASRI, Department of ECE, Seoul National University
{ghimhw, dsybaik, cms6539, ultio791, kyoungmu}@snu.ac.kr

Abstract

We present a novel framework for controllable image restoration that can effectively restore multiple types and levels of degradation of a corrupted image. The proposed model, named TASNet, is automatically determined by our neural architecture search algorithm, which optimizes the efficiency-accuracy trade-off of the candidate model architectures. Specifically, we allow TASNet to share the early layers across different restoration tasks and adaptively adjust the remaining layers with respect to each task. The shared task-agnostic layers greatly improve the efficiency while the task-specific layers are optimized for restoration quality, and our search algorithm seeks for the best balance between the two. We also propose a new data sampling strategy to further improve the overall restoration performance. As a result, TASNet achieves significantly faster GPU latency and lower FLOPs compared to the existing state-of-the-art models, while also showing visually more pleasing outputs. The source code and pre-trained models are available at https://github.com/ghimhw/TASNet.

1. Introduction

Restoration of real-world corrupted images is a challenging problem since the types and the severity (or level) of degradation are unknown. Previous works on *blind* image super-resolution [4, 26] or blind deblurring [35, 14, 1] tackle this problem by learning to predict the unknown degradation kernel, and then using the predicted kernel to restore clean images. Recently, *controllable* image restoration has been gaining increased attention as alternative approaches. In this scenario, instead of accepting a single restored image given by the final model, users can control the output restoration to generate multiple images and choose the output image that best fits their preferences.

Early works on controllable image restoration (CIR) [15, 27, 31, 32] mostly consider a single type of degradation and modulate the levels of restoration. For instance, the denoising model from [15] allows continuous modulation of denoising a Gaussian noise with $\sigma = 15 \sim 75$. More recently, CResMD [16] proposed an extended framework that learns multiple types of degradation (Gaussian blur, Gaussian noise, and JPEG compression) jointly with a single network, so that users can interactively adjust not only the level but also the type of degradation. However, as more flexible control is enabled, two new challenges arise for the practical application of CIR models: 1) the high computation cost of generating multiple images to choose from, and 2) the difficulty of finding the true types and the levels of degradation, in which failing to do so may lead to significantly deteriorated outputs.

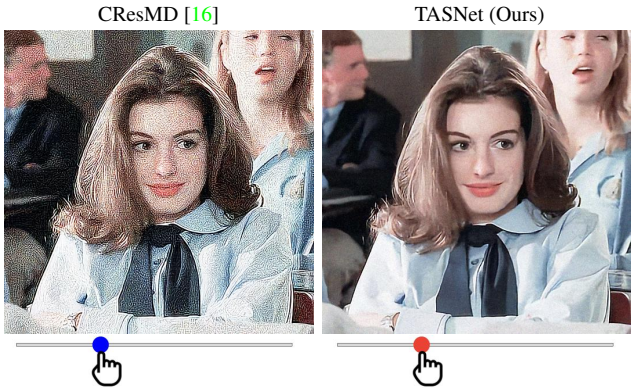


Figure 1: **An example of controllable image restoration**. Our model generates visually more pleasing outputs while adjusting restoration levels with **3 times faster GPU latency** and **95.7% reduced FLOPs** compared to CResMD [16].

To alleviate these limitations, we present **TASNet**, a novel deep-learning-based CIR model that is optimized to achieve better visual quality and substantially reduced computational complexity. Figure 1 demonstrates a sample result. Our TASNet consists of two parts: task-agnostic layers and task-specific layers, where we denote "task" as a restoration problem *w.r.t.* a combination of degradation types and levels. The task-agnostic part is composed of the early layers of the baseline supernetwork, where the parameters of the layers are shared across all tasks. Sharing the early layers greatly improves the efficiency of CIR model, since we do not need to re-compute the output of the shared layers each time a

不同输出 share一些 layer.

*Now at Google Research
†Now at Kookmin University

Figure 2: The overview of our efficient architecture for controllable image restoration. (a) CResMD [16] has a fixed network across all tasks and requires separate inference through the full model whenever the target restoration task becomes different. (b) Our task-agnostic and task-specific network (TASNet) shares the early layers to facilitate feature reuse. When we perform inference for multiple tasks, the task-agnostic part requires only a single computation, of which the output feature can be reused multiple times as the input for the task-specific network. The architecture of the task-specific network is adaptively adjusted for each given task. The width and the height of boxes represent the number of layers and channels of neural networks, respectively. In this example, two popular restoration tasks of denoising and deblurring are visualized, where different colors represent the corresponding inference path.

user changes the task (the type or the level of degradation). On the other hand, the remaining layers that consist of the task-specific parts are differently adjusted for each task. The main concept is summarized in Figure 2. However, deciding the architectural hyperparameters that balances the efficiency and the performance is still a very challenging problem.

To this end, we propose a new supernetwork-based neural architecture search (NAS) algorithm that can automatically search for the task-agnostic and task-specific architectures on the efficiency-accuracy trade-off curve. Since we need to consider a large number of tasks for continuously varying levels of restoration, the search space of our algorithm should be able to represent a diverse set of architectures. This is why our algorithm allows channel-level selection for each layer as well as layer-wise decision of whether to share its parameters or not. Specifically, the proposed NAS algorithm selects: 1) the number of layers to share (task-agnostic part), 2) the important channels for the shared layers, and 3) the important channels for each task-specific layer, where these task-specific channel selection is adaptive for each task. We also formulate the overall learning objective to be differentiable for efficient end-to-end training of our searching framework, which results in the final TASNet. Moreover, we propose a new data sampling strategy to reduce the visual artifacts, which is empirically shown to be effective for cases when the task given by the user is very different from the true degradation of an input image.

Experimental results show that TASNet runs 3.7 times faster than the state-of-the-art CIR model on modern high-end GPUs with 95.7% FLOPs reduction when generating 4K images with 27 modulations. Also, the visual quality of the generated restoration using TASNet is much better than the previous approaches with significantly less artifacts.

Overall, our contributions can be summarized as follows:

- We present a novel neural network, named TASNet, for controllable image restoration (CIR) that remarkably improves the model efficiency and output image quality.
- We propose a supernetwork-based NAS algorithm that finds efficient CIR networks in a differentiable manner.
- We introduce a new data sampling strategy to improve the generated image quality in CIR problems.
- The proposed TASNet outperforms the state-of-the-art models in image quality and computation costs of FLOPs and CPU/GPU latency.

2. Related Work

2.1. Image restoration

Image restoration, including denoising, deblurring, super-resolution, and compression artifact removal, is a classical topic in computer vision which aims at restoring the original image from its degraded versions. Deep-learning-based image restoration networks [10, 11, 12, 18, 19, 21, 40, 42] have achieved breakthroughs in restoring accurate image details. While the conventional approaches specialize in dealing with a single degradation type, general image restoration aims to restore the corrupted image with multiple types of degradation. Notable approaches include learning a policy to determine a specialized restoration network for the input image [36, 37], or using an operation-wise attention module to produce the specialized feature maps *w.r.t.* each degradation type [29]. However, these approaches cannot control the diverse restoration levels for the input images, and sometimes generate overly smooth or sharpened outputs.

On the other hand, controllable image restoration is recently gaining interests from the computer vision research community, to control the output restoration of an image

corrupted by unknown degradation. Existing works learn to control restoration levels for a single type of degradation [15, 27, 31, 32]. In particular, AdaFM [15], CFSNet [31], and Dynamic-Net [27] design their network architectures with tuning modules, which modulate the feature maps layer-wise [15] or block-wise [31, 27] with respect to the tasks of interest at test time. Instead, DNI [32] interpolates all parameters of the differently trained networks for modulation. For the general controllable image restoration, CResMD [16] controls restoration levels in multiple types of degradation with a block-wise tuning module. While the prior works may have provided good performance to control the restoration levels, they have solely focused on the image quality and do not consider computational efficiency. By contrast, using CResMD as the baseline, the proposed TASNet significantly reduces the computations and running time.

本质是对 CResMD 的改进?

2.2. Efficient CNNs for image restoration

To make the image restoration models efficient with less computation cost, several novel architectures have been developed for diverse restoration tasks. The early works downscale the spatial resolution of the input image to reduce the computation costs of the convolution operations for denoising [41] and super-resolution [12]. More recently, CARN [3] presents a cascading residual block with multiple skip connections, leading to a fast and light-weight super-resolution network. For deblurring, a method using spatially variant deconvolution is proposed in [38] to achieve accurate performance with its efficient backbone network. Meanwhile, FALSR [8], ESRN [28], and FGNAS [17] adopt neural architecture search (NAS) algorithms for efficient super-resolution networks. Path-Restore [37] and AdaDSR [23] save computation costs via adaptive inference for general image restoration and super-resolution, respectively. Prior works also employ network quantization [9, 34] or pruning [13, 20, 24, 30, 39], but they are not task-adaptive.

On the other hand, we study the network acceleration approaches for controllable image restoration for the first time, especially when it requires a large number of inference passes per image. A neural architecture accelerated by our algorithm can be considered as a special instance of multi-task learning [7, 43], a network design paradigm that uses a shared network for multiple tasks or optimization. The main difference from the previous multi-task learning approaches is employing NAS for continuously varying tasks from an input image. Our search algorithm is a variant of supernetwork-based NAS methods [17, 22], which aim to find an efficient or performance-wise optimal network by pruning from a supernetwork. Our search process is performed over a search space of channels and shared layers across tasks, each combination of which provides a candidate network derived from a supernetwork.

3. Method

Controllable image restoration (or modulation) aims to control the restoration levels of a corrupted image. Following the CResMD setting, we formulate multi-dimensional restoration levels to be controllable. Formally, given D number of degradation types, $\boldsymbol{t} \in \mathbb{R}^D$ denotes a task vector, where $t_d \in [0, 1]$ encodes the restoration level for the d-th degradation type. For instance, a task vector of (1,0,0) for three degradation types (*e.g.*, blur, noise, JPEG compression) indicates that the task requires the maximum level of deblurring but no denoising or compression artifact removal. During training, a training image pair (input and target) determines the corresponding values of task vector. At inference time, the task vector values are controlled by the users.

3.1. Efficient architecture design

训的时候知道具体 degradation，推理前提.

Unknown degradation of real images demands interactive image restoration with adjustable restoration levels. In this scenario, a network for image modulation computes its operations multiple times for a single input image with different task vectors. Formally, the total computation cost for M times of inferences is given by,

$$\mathcal{R}_{\text{total}}(f, \boldsymbol{x}, \boldsymbol{t}) = \sum_{m=1}^{M} \mathcal{R}(f, \boldsymbol{x}, \boldsymbol{t}_m), \tag{1}$$

where $\mathcal{R}(f, \boldsymbol{x}, \boldsymbol{t}_m)$ denotes FLOPs or latency to generate an output with the network architecture f, the input image $\boldsymbol{x}$, and the m-th task vector $\boldsymbol{t}_m$. Architectures used in CResMD and other previous works [15, 27, 31, 32] follow the computation cost of Equation (1), as outlined in Figure 2(a).

Our goal is to design a network architecture which is efficient under the aforementioned multiple -inference scenario. To this end, we propose **TASNet** that shares the feature map of early layers with the remaining task-specific architecture, as described in Figure 2(b). The task-agnostic shared layers facilitate feature reuse for repeated inferences from a single image. On the other hand, our task-specific architecture *adaptively* transforms itself to be efficient as it is difficult to find a single fixed network that is efficient for continuously varying restoration levels.

For TASNet, we reformulate Equation (1) and divide the network f into the early layers f^a and the remaining layers f^s. Then, the total computation cost becomes:

$$\begin{aligned} \mathcal{R}_{\text{total}}(f, \boldsymbol{x}, \boldsymbol{t}) &= \sum_{m=1}^{M} \left[\mathcal{R}(f^a, \boldsymbol{x}) + \mathcal{R}(f^s, \tilde{\boldsymbol{x}}, \boldsymbol{t}_m)\right] \\ &\geq \mathcal{R}(f^a, \boldsymbol{x}) + \sum_{m=1}^{M} \mathcal{R}(f^s_m, \tilde{\boldsymbol{x}}, \boldsymbol{t}_m), \end{aligned} \tag{2}$$

where $\tilde{\boldsymbol{x}} = f^a(\boldsymbol{x})$ and $\mathcal{R}(f^a, \boldsymbol{x})$ is the computation cost of *a single inference* for $f^a(\boldsymbol{x})$, and f_m denotes the transformed

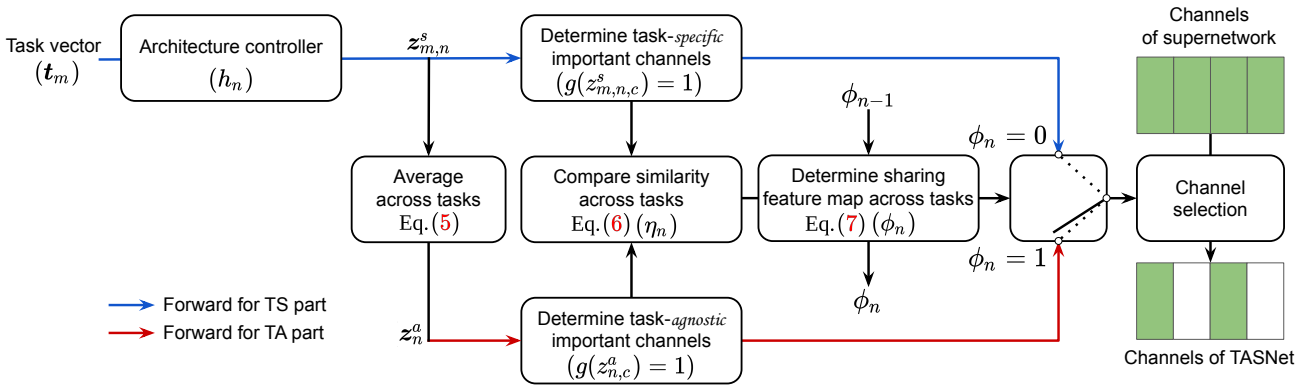


Figure 3: The neural architecture search process for each layer of TASNet. Our algorithm automatically determines the number of shared layers and channels in each feature map from the supernetwork. Task-specific (TS) part ($\phi_n = 0$) adaptively selects channels based on the given task (**blue arrow**). By contrast, task-agnostic (TA) part ($\phi_n = 1$) selects fixed channels across tasks (**red arrow**). A feature map is determined to be shared if the channel importance is similar among tasks and the previous feature map is shared. All processes are differentiable via a straight-through estimator (g). During the inference, ϕ, $\boldsymbol{z}^a$, and thus task-agnostic (TA) part are fixed.

task-specific architecture. Although Equation (2) should theoretically reduce the computational redundancy, designing efficient architectures (f^a and f^s_m) is still an open problem.

3.2. Search formulation

Overview. In order to find efficient TASNet architectures, we propose a supernetwork-based neural architecture search algorithm. Our search algorithm determines 1) the number of early layers that are shared across tasks, 2) the important channels for task-agnostic layers, and 3) the important channels for each task-specific layer, where the channels are selected from the supernetwork CResMD [16]. TASNet aims to minimize both restoration error and computation cost of Equation (2) via following rules, as illustrated in Figure 3:

- Learn task-specific channel importance ($\boldsymbol{z}^s_m$).
- Learn task-agnostic channel importance ($\boldsymbol{z}^a$).
- Share a feature map across tasks ($\phi_n = 1$), when important channels are similar across tasks ($\eta_n = 1$) and the feature map of its previous layer is shared ($\phi_{n-1} = 1$).
- Maximize the number of shared layers (Equation (10)).
- Prune unimportant channels across tasks ($g(z^a_{n,c}) = 0$) in shared feature maps ($\phi_n = 1$).
- Adaptively select important channels ($g(z^s_{m,n,c}) = 1$) to the task $\boldsymbol{t}_m$ in non-shared feature maps ($\phi_n = 0$).

Channel selection. Variants of straight-through estimator [5] have been widely used for differentiable NAS approaches [33, 6]. To *select* or *de-select* each channel from the super network, channel selection virtually multiplies a binary value to the channel. Our straight-through estimator enables this process differentiable, formally given by,

$$g(z) = \begin{cases} \mathbb{I}[z > 0] & \text{if forward} \\ \text{sigmoid}(z) & \text{if backward,} \end{cases} \tag{3}$$

where $z \in \mathbb{R}$, and $\mathbb{I}[\cdot]$ is an indicator function that returns 1 when its input is true (and 0 otherwise). We introduce two types of z which determine task-specific and task-agnostic channels, respectively, in the following.

Task-specific channel importance. To learn channel importance for a given task $\boldsymbol{t}_m$, we introduce architecture controller $\boldsymbol{h}$, formally given by,

$$\boldsymbol{z}^s_{m,n} \equiv h_n(\boldsymbol{t}_m), \tag{4}$$

where $z^s_{m,n,c} \in \mathbb{R}$ denotes the importance of c-th channel to the task vector $\boldsymbol{t}_m$ in the n-th feature map of the supernetwork. h_n is the architecture controller, composed of few fully connected layers, for the n-th feature map.

Task-agnostic channel importance. To learn general channel importance across tasks, we simply average the values of the task-specific channel importance as follows:

$$z^a_{n,c} \equiv \frac{1}{M} \cdot \sum_{m=1}^{M} z^s_{m,n,c}, \tag{5}$$

where $z^a_{n,c} \in \mathbb{R}$ denotes the task-agnostic channel importance and M is a large enough number of inference. Empirically, we adopt exponential moving average over iterations with the small mini-batch size.

Channel importance similarity across tasks. To determine whether a feature map should be shared across tasks, we compute the agreement criterion via the similarity between selected channels from $\boldsymbol{z}^a$ and $\boldsymbol{z}^s$ as follows:

$$\frac{1}{M} \cdot \sum_{m=1}^{M} \sum_{c=1}^{C} g(z^s_{m,n,c}) \cdot g(z^a_{n,c}) > \gamma \cdot \sum_{c=1}^{C} g(z^a_{n,c}), \tag{6}$$

where γ is a threshold hyperparameter. Whether Equation (6) holds is represented by a boolean variable η_n. If the equation

holds ($\eta_n = 1$), a large number of tasks have an agreement on the channel importance for a given layer, and thus this layer is likely to be shared across all tasks.

Recursive layer sharing. To facilitate feature reuse across tasks, the shared layers need to be located together at the initial stage of the network. To this end, the n-th feature map is shared if the n-th and all previous feature maps satisfy the agreement criterion on the position of pruning across tasks ($\eta_i = 1$) , formally given by,

$$\phi_n = \begin{cases} 1 & \text{if } \eta_i = 1, \forall i = 1, 2, ..., n \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

where $\boldsymbol{\phi} \in \mathbb{Z}_2^N$ denotes a decision variable, in which the n-th element ϕ_n is 1 if the n-th feature map is task-agnostic.

Objective function. By using all equations above, we can formulate the objective function with differentiable resource regularization terms. Let $\mathcal{L}(\cdot, \cdot)$ denote a standard ℓ_1 loss function for image restoration tasks. The overall objective function is formally given by,

$$\min_{\theta, \psi} \mathcal{L}(\theta, \psi) + \lambda_1 \cdot \mathcal{R}_1(\psi) + \lambda_2 \cdot \mathcal{R}_2(\psi), \tag{8}$$

where θ and ψ are learnable parameters in the supernetwork and architecture controller, respectively. The first resource regularizer $\mathcal{R}_1(\cdot)$ penalizes FLOPs of currently searched architectures by *de-selecting* channels, formally defined as:

$$\begin{aligned} \mathcal{R}_1(\psi) &= \mathcal{R}_{\text{FLOPs}}(f^a, \boldsymbol{x}) + \sum_{m=1}^{M} \mathcal{R}_{\text{FLOPs}}(f^s, \tilde{\boldsymbol{x}}, \boldsymbol{t}_m) \\ &= 2 \sum_{n=1}^{N} \mathrm{K}_n^2 H_n W_n \cdot [\phi_n \cdot \sum_{c=1}^{C} g(z_{n,c}^a) \cdot \sum_{c=1}^{C} g(z_{n-1,c}^a) \\ &\quad + (1 - \phi_n) \cdot \sum_{m=1}^{M} \{ \sum_{c=1}^{C} g(z_{m,n,c}^s) \cdot \sum_{c=1}^{C} g(z_{m,n-1,c}^s) \}], \end{aligned} \tag{9}$$

where K_n is the kernel size of convolution operation to generate the n-th feature map, H_n and W_n are the height and the width of the n-th feature map, respectively, and $z_{0,c}^a$ and $z_{m,0,c}^s$ the channel of input images and are fixed to be 1. The second regularizer $\mathcal{R}_2$ enforces the network to maximize the number of the early shared layers by penalizing the *disagreement* of selected channels across tasks as follows:

$$\mathcal{R}_2(\psi) = \sum_{n=1}^{N} \phi_{n-1} \cdot \sum_{c=1}^{C} \sum_{m=1}^{M} \left\| g(z_{m,n,c}^s) - g(z_{n,c}^a) \right\|_1, \tag{10}$$

where layer at $n = 0$ denotes an input image and $\phi_0 \equiv 1$ since the input image is fixed over tasks. The hyperparameters λ_1 and λ_2 balance three terms.

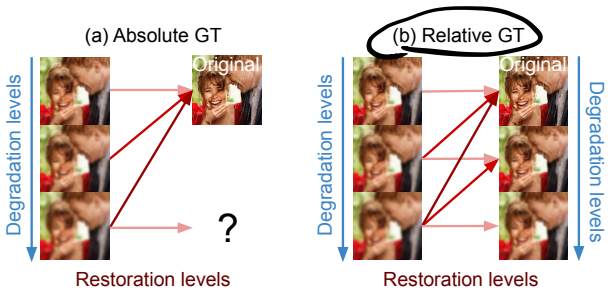


Figure 4: Absolute ground truth *vs.* relative ground truth. (a) Mapping from all degraded versions to its original image. (b) Mapping from degraded versions to relatively higher-quality images.

3.3. Data sampling strategy

Degradation level *vs.* restoration level. Previous works train a network to restore the original image from the degraded images with arbitrary degradation level (see Figure 4(a)). However, CIR algorithms should be able to restore images to various extents to facilitate better user interaction experience. Thus, we redefine a restoration level as a mapping from more degraded images (input) to less degraded images (relative GT) (see Figure 4(b)).

Task vector with relative GT. A task vector $\boldsymbol{t}$ is a model input that encodes restoration levels. In training, an input-GT image pair (sampled with two different multi-dimensional degradation levels) determines its task vector as follows:

$$t_d \equiv l_d^{in} - l_d^{gt}, \tag{11}$$

where $l_d^{in} \in [0, 1]$ and $l_d^{gt} \in [0, 1]$ denote the levels of d-th degradation type for the input and GT images, respectively. We assume GT images are less degraded than input ($l_d^{in} \geq l_d^{gt}$). Each training image pair randomly selects the number of degradation types (single or multiple) and the granularity of degradation levels (continuous or binary).

4. Experiments

In this section, we present the experimental results and comparisons between TASNet and CResMD in terms of network computation cost and output image quality. Then, we thoroughly analyze the effectiveness of our proposed algorithm with the ablation studies. Implementation details are described in the supplementary document.

4.1. Dataset

In this work, we use DIV2K [2] dataset for training and CBSD68 [25] dataset for testing, unless specified otherwise. DIV2K consists of 800 clean 2K-resolution training images and 100 validation images while CBSD68 consists of 68 clean HVGA-resolution test images. Following the degradation setting in CResMD [16], to create degraded images, we use three types of degradation: Gaussian blur, Gaussian

Table 1: Comparisons on the average computation cost. TASNet outperforms CResMD [16] *w.r.t.* all measures and resolutions.

Cost metric	Resolution	CResMD	TASNet
FLOPs$_\downarrow$	HD	1,124.3 G	**45.2 G**
	2K	2,698.4 G	**108.4 G**
	4K	10,119.2 G	**406.7 G**
CPU latency (single)$_\downarrow$	HD	22.8 s	**5.5 s**
	2K	55.6 s	**13.5 s**
	4K	209.3 s	**55.5 s**
CPU latency (multi)$_\downarrow$	HD	5.1 s	**1.7 s**
	2K	11.7 s	**4.2 s**
	4K	40.6 s	**13.1 s**
GPU latency$_\downarrow$	HD	144.4 ms	**68.4 ms**
	2K	280.8 ms	**99.2 ms**
	4K	930.0 ms	**250.7 ms**

Table 2: Non-blind quantitative image quality results on CBSD68.

Method	PSNR$_\uparrow$	SSIM$_\uparrow$	NIQE$_\downarrow$	BRISQUE$_\downarrow$	FLOPs$_\downarrow$
CResMD	**25.86 dB**	**0.8194**	6.7165	54.13	189.1 G
TASNet	25.64 dB	0.8137	**6.6301**	**50.60**	**7.5 G**

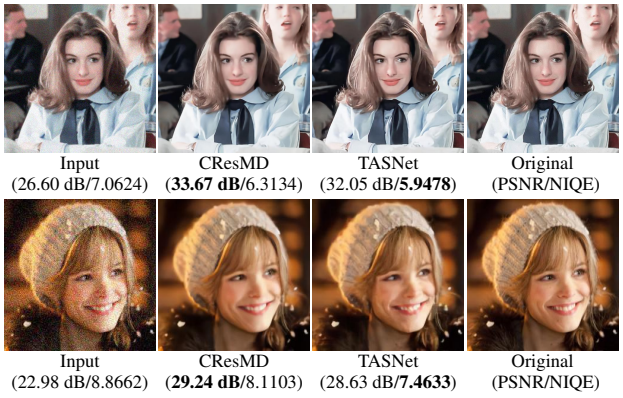


Figure 5: Non-blind qualitative image quality comparison. TASNet produces sharper images with better NIQE scores than CResMD.

noise, and JPEG compression. Each degradation is sequentially applied to the clean images. For Gaussian blur, we use the kernel size of 21 $\times$ 21 with the radius $r \in [0, 4]$. The covariance for the Gaussian noise is denoted as $\sigma \in [0, 50]$. The JPEG compression quality factor is denoted as $q \in [10, 100]$ (We also include images with no JPEG compression as in CResMD). The training dataset is constructed by applying the degradation levels with a stride of 0.1, 1, and 2 for r, σ, and q, respectively.

4.2. Computation cost comparison

Latency and FLOPs reduction. Table 1 presents the average computation cost of TASNet (ours) and the state-of-the-art network, CResMD [16], across diverse image resolutions and devices. The experiments are performed for the controllable image restoration setting, in which a multiple number ($M = 27$) of inferences are performed for each input image.

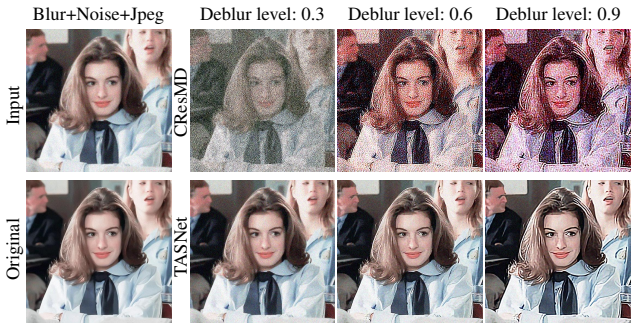


Figure 6: **Blind setting (artifact)**. CResMD often produces significant artifacts when deburring images that are corrupted with noise or compression artifacts. By contrast, TASNet successfully reduces the blur in input images.

While displaying similar image restoration performance (as described in the next section), TASNet manages to reduce 95.7% FLOPs from CResMD and shows faster speed on all reported devices: $\times$3.8 on a single-core CPU, $\times$3.1 on a multi-core CPU, and $\times$3.7 on a GPU, when generating 4K (3840$\times$2160) images, compared with CResMD. Notably, TASNet only requires 0.07s to restore an HD (1280$\times$720) image. We also observe that the latency difference between two models becomes small in the case of low-resolution images. As the input resolution decreases, the size of each feature map also decreases, reducing the benefit of selecting channels or shared layers to some extent.

4.3. Image quality comparison

Non-blind setting. Table 2 illustrates the quantitative image quality comparisons in a non-blind image restoration setting, where the degradation type and level of input images are known. This setting allows models to generate their best results with a single inference. The results demonstrate that the images restored by TASNet have lower PSNR than the images generated by CResMD but better NIQE, which means that TASNet in general restores sharper image details, as illustrated in Figure 5.

Blind setting. Controllable image restoration (CIR) algorithms aim to tackle a blind setting, in which the types and levels of degradation are unknown. CResMD [16] struggles to handle such challenging scenario, generating images with artifact (Figure 6) or over-smoothing effect (Figure 7) and restoring images unevenly across continuously varying restoration levels (Figure 8). Figure 6 presents images restored by algorithms that modulate an input image using different levels of deblurring when the input image is corrupted with a mixture of blur, noise, and compression degradation. Images restored by TASNet are shown to be less blurry (in fact, the outputs become sharper as deblurring level becomes higher), compared to CResMD that generates critical artifacts. An interesting observation is that other degradations

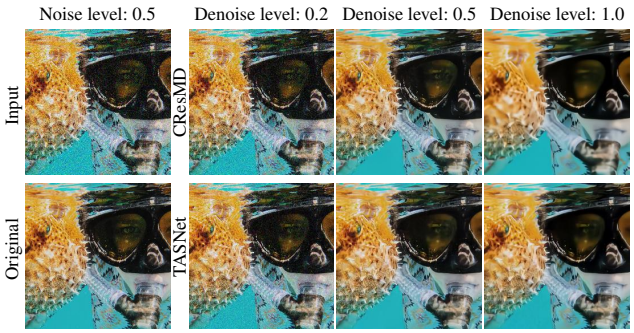


Figure 7: **Blind setting (over-smoothing)**. When denoise levels are higher than the actual noise levels of input images, CResMD over-smoothes images whereas TASNet noticeably removes noise.

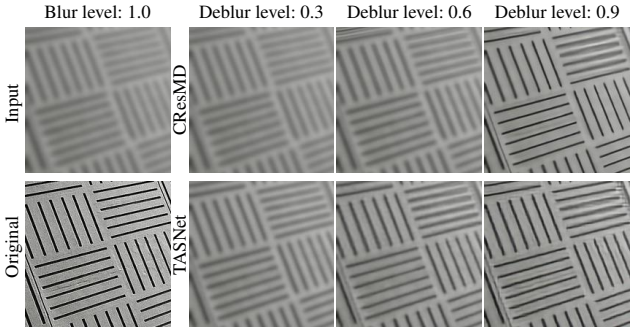


Figure 8: **Blind setting (uneven modulation)**. CResMD restores blurred images negligibly or drastically by deblur level changes, while TASNet gradually modulates outputs.

(noise and JPEG artifact) still remain in images deblurred by TASNet, implying that our algorithm manages to learn a disentangled restoration process for each restoration type and level. As observed in Figure 7, denoising by CResMD results in over-smooth images while TASNet maintains the overall contents and structures of input images. Figure 8 illustrates the restored images when varying the restoration (deblurring in this case) levels for identical degradation and restoration types. With the same amount of change in restoration levels, CResMD restores the degradation unevenly (the restoration quality changes negligibly or drastically), whereas TASNet generates images with smoothly-varying restoration quality.

Restoration on real images. Figure 9 displays the output images restored from real images with unknown degradation (downloaded from the internet). Similar to the synthetic examples, CResMD generates images with over-smooth effect or significant artifacts while TASNet successfully reduces the degradation of input images. For more restored output images, please refer to the supplementary document.

4.4. Ablation study

The effectiveness of sharing layers in TASNet. Table 3 studies the importance of task-agnostic layers in TASNet. In

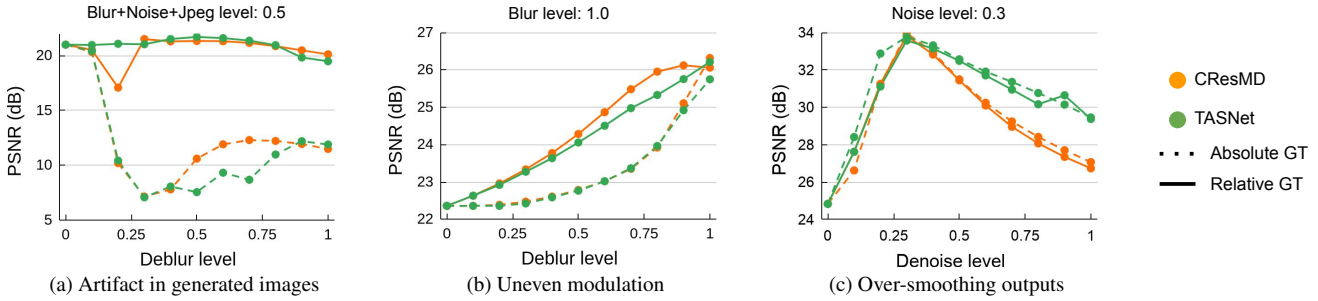
Figure 9: **Restoration from real images**. A task vector $(\cdot,\cdot,\cdot)$ denotes the levels of (Deblur, Denoise, Dejpeg). Compared to TASNet, CResMD generates more artifacts or over-smoothed images.

Table 3: Ablation study for the effectiveness of shared layers. TA and TS, respectively, denote task-agnostic layers and task-specific layers. TSNet is our searched model without forcibly sharing layers. Image quality is measured on CBSD68 using the non-blind setting.

Method	TA	TS	PSNR$_\uparrow$	NIQE$_\downarrow$	FLOPs$_\downarrow$
CResMD	-	-	**25.86 dB**	6.7165	189.1 G
TSNet	-	✓	25.59 dB	6.6332	39.6 G
TASNet	✓	✓	25.64 dB	**6.6301**	**7.5 G**

particular, we examine how the performance and computation cost change after disabling layer sharing, the resulting network from which is denoted as TSNet in the table. TASNet is observed to save the computation costs of TSNet by more than 5 times, owing to its shared layers that allow feature reuse and thus reducing the redundant computation for multiple inferences. Regardless, TSNet greatly reduces the computation cost of CResMD, suggesting the effectiveness of the task-specific layers that adaptively select important channels. In stark contrast, conventional channel pruning approaches do not change their architectures *w.r.t.* the task.

To further emphasize the effectiveness of TASNet, Figure 11 shows the computation cost of TSNet and TASNet across various tasks with multiple number (4) of inferences for each image of HVGA resolution (481×321). Task-specific layers in both networks tend to require more channels for higher restoration levels, which translate to more difficult restoration problems. Task-agnostic layer in



(a) Artifact in generated images (b) Uneven modulation (c) Over-smoothing outputs

Figure 10: Ablation study for three image modulation problems in the blind setting. Models trained by relative GT reduce (a) network artifact generation when deblurring images corrupted by a mixture of degradation types and (b) uneven image modulation across deblurring levels. Further, task-agnostic feature maps from TASNet prevent (c) over-smoothing outputs.

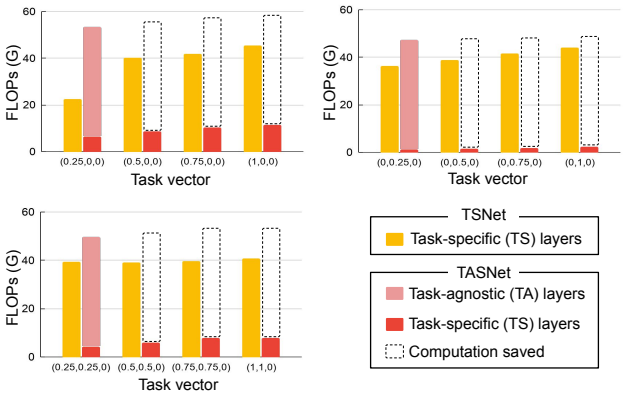


Figure 11: Computation cost comparisons across restoration levels. The higher restoration levels demand more computation costs for TS layers. TASNet is efficient for multiple inferences, as a result of reusing the feature map of TA layers across tasks. Each graph presents computation costs of removing blur (upper left), noise (upper right), and joint degradation of blur and noise (lower left). A task vector $(\cdot,\cdot,\cdot)$ denotes the levels of (Deblur, Denoise, Dejpeg).

TASNet is computed only once for each input image and hence requires a substantially smaller amount of computation from the second pass. Although TASNet requires higher computation cost than TSNet for the first inference, the overhead becomes negligible during multiple inferences.

Comparison to naïve shared networks. Table 4 shows the comparisons between the variations of CResMD with a different number of early shared layers that is manually determined. For fair comparisons in terms of performance, all models in the table are trained with absolute GT. TASNet-A has the same network architecture as TASNet, but is trained with absolute GT. TASNet-A reduces the computation cost of CResMD to $\frac{1}{9}$ by sharing 62% of the layers while maintaining the similar PSNR performance.

Image restoration quality analysis. To study the effectiveness of the proposed data sampling strategy and the TASNet architecture, Figure 10 presents quantitative restoration performance of three major failure cases in CIR when the controlled restoration levels differ from the actual types and

Table 4: Ablation study of naïve shared networks. We modify CResMD by sharing its early layers. TASNet-A achieves 9 times FLOPs reduction than CResMD with 62% shared layers.

Method	#Shared Layer	PSNR$_\uparrow$	NIQE$_\downarrow$	FLOPs$_\downarrow$
CResMD	0 %	25.86 dB	6.7165	189.1 G
	31 %	25.82 dB	6.8035	132.0 G
	62 %	25.78 dB	6.8205	69.5 G
	99 %	25.34 dB	6.9109	7.0 G
TASNet-A	62 %	25.75 dB	6.7982	7.5 G

levels of degradation in a corrupted image. The results of CResMD trained with relative GT generates less artifacts and evenly modulated outputs, validating the capability of the proposed data sampling to improve image restoration quality. Also, TASNet achieves higher PSNR (over 3 dB compared to CResMD) in denoising without over-smoothing (Figure 7 and 10(c)), alluding to the effectiveness of the shared layers in providing better image quality.

5. Conclusion

We propose a novel neural architecture search algorithm to find efficient networks for controllable image restoration (or image modulation). In particular, the proposed algorithm searches for a network with task-agnostic and task-specific layers, referred to as TASNet, by determining the number of layers and channels to share across tasks and adaptively selecting channels in non-shared feature maps. We formulate all learning objectives in a differentiable manner and perform the architecture search in an end-to-end manner. The shared layers facilitate feature reuse that pushes the network efficiency further for controllable image restoration that requires a several number of inferences. Together with the proposed new data sampling strategy, not only does TASNet reduce the network computation costs of the state-of-the-art network greatly but also provides the better image quality.

Acknowledgment This work was supported in part by an IITP grant funded by the Korean government [No. 2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University)].

References

[1] Raanan Fattal Adam Kaufman. Deblurring using analysis-synthesis networks pair. In *CVPR*, 2020. 1

[2] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *CVPRW*, 2017. 5

[3] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *ECCV*, 2018. 3

[4] Sefi Bell-Kligler, Assaf Shocher, and Michal Irani. Blind super-resolution kernel estimation using an internal-gan. In *NeurIPS*, 2019. 1

[5] Yoshua Bengio. Estimating or propagating gradients through stochastic neurons. *arXiv preprint arXiv:1305.2982*, 2013. 4

[6] Han Cai, Ligeng Zhu, and Song Han. ProxylessNAS: Direct neural architecture search on target task and hardware. In *ICLR*, 2019. 4

[7] Rich Caruana. Multitask learning. In *MLJ*, 1997. 3

[8] Xiangxiang Chu, Bo Zhang, Hailong Ma, Ruijun Xu, Jixiang Li, and Qingyuan Li. Fast, accurate and lightweight super-resolution with neural architecture search. In *ICPR*, 2020. 3

[9] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. In *NIPS*, 2016. 3

[10] Chao Dong, Yubin Deng, Chen Chang Loy, and Xiaoou Tang. Compression artifacts reduction by a deep convolutional network. In *ICCV*, 2015. 2

[11] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *TPAMI*, 2014. 2

[12] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *ECCV*, 2016. 2, 3

[13] Yonggan Fu, Wuyang Chen, Haotao Wang, Haoran Li, Yingyan Lin, and Zhangyang Wang. Autogan-distiller: Searching to compress generative adversarial networks. In *ICML*, 2020. 3

[14] Dong Gong, Jie Yang, Lingqiao Liu, Yanning Zhang, Ian Reid, Chunhua Shen, Anton van den Hengel, and Qinfeng Shi. From motion blur to motion flow: A deep learning solution for removing heterogeneous motion blur. In *CVPR*, 2017. 1

[15] Jingwen He, Chao Dong, and Yu Qiao. Modulating image restoration with continual levels via adaptive feature modification layers. In *CVPR*, 2019. 1, 3

[16] Jingwen He, Chao Dong, and Yu Qiao. Interactive multi-dimension modulation with dynamic controllable residual learning for image restoration. In *ECCV*, 2020. 1, 2, 3, 4, 5, 6

[17] Heewon Kim, Seokil Hong, Bohyung Han, Heesoo Myeong, and Kyoung Mu Lee. Fine-grained neural architecture search. *arXiv preprint arXiv:1911.07478*, 2019. 3

[18] Jiwon Kim, Jungkwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016. 2

[19] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017. 2

[20] Yawei Li, Shuhang Gu, Kai Zhang, Luc Van Gool, and Radu Timofte. Dhp: Differentiable meta pruning via hypernetworks. In *ECCV*, 2020. 3

[21] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, 2017. 2

[22] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *ICLR*, 2019. 3

[23] Ming Liu, Zhilu Zhang, Liya Hou, Wangmeng Zuo, and Lei Zhang. Deep adaptive inference networks for single image super-resolution. In *ECCVW*, 2020. 3

[24] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *ICCV*, 2017. 3

[25] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001. 5

[26] Assaf Shocher, Nadav Cohen, and Michal Irani. "zero-shot" super-resolution using deep internal learning. In *CVPR*, 2018. 1

[27] Alon Shoshan, Roey Mechrez, and Lihi Zelnik-Manor. Dynamic-net: Tuning the objective without re-training for synthesis tasks. In *ICCV*, 2019. 1, 3

[28] Dehua Song, Chang Xu, Xu Jia, Yiyi Chen, Chunjing Xu, and Yunhe Wang. Efficient residual dense block search for image super-resolution. In *AAAI*, 2020. 3

[29] Masanori Suganuma, Xing Liu, and Takayuki Okatani. Attention-based adaptive selection of operations for image restoration in the presence of unknown combined distortions. In *CVPR*, 2019. 2

[30] Haotao Wang, Shupeng Gui, Haichuan Yang, Ji Liu, and Zhangyang Wang1. Gan slimming: All-in-one gan compression by a unified optimization framework. In *ECCV*, 2020. 3

[31] Wei Wang, Ruiming Guo, Yapeng Tian3, and Wenming Yang. Cfsnet: Toward a controllable feature space for image restoration. In *ICCV*, 2019. 1, 3

[32] Xintao Wang, Ke Yu, Chao Dong, Xiaoou Tang, and Chen Change Loy. Deep network interpolation for continuous imagery effect transition. In *CVPR*, 2019. 1, 3

[33] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *CVPR*, 2019. 4

[34] Jingwei Xin, Nannan Wang, Xinrui JiangJie Li, Heng Huang, and Xinbo Gao. Binarized neural network for single image super resolution. In *ECCV*, 2020. 3

[35] Xiangyu Xu, Jinshan Pan, Yujin Zhang, and Ming-Hsuan Yang. Motion blur kernel estimation via deep learning. *TIP*, 2018. 1

[36] Ke Yu, Chao Dong, Liang Lin, and Chen Change Loy. Crafting a toolchain for image restoration by deep reinforcement learning. In *CVPR*, 2018. 2

[37] Ke Yu, Xintao Wang, Chao Dong, Xiaoou Tang, and Chen Change Loy. Path-restore: Learning network path selection for image restoration. *arXiv preprint arXiv:1904.10343*, 2019. 2, 3

[38] Yuan Yuan, Wei Su, and Dandan Ma. Efficient dynamic scene deblurring using spatially variant deconvolution network with optical flow guided training. In *CVPR*, 2020. 3

[39] Sun-Yuan Kung Zejiang Hou. Efficient image super resolution via channel discriminative deep neural network pruning. In *ICASSP*, 2020. 3

[40] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *TIP*, 2017. 2

[41] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for CNN based image denoising. *TIP*, 2018. 3

[42] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *CVPR*, 2018. 2

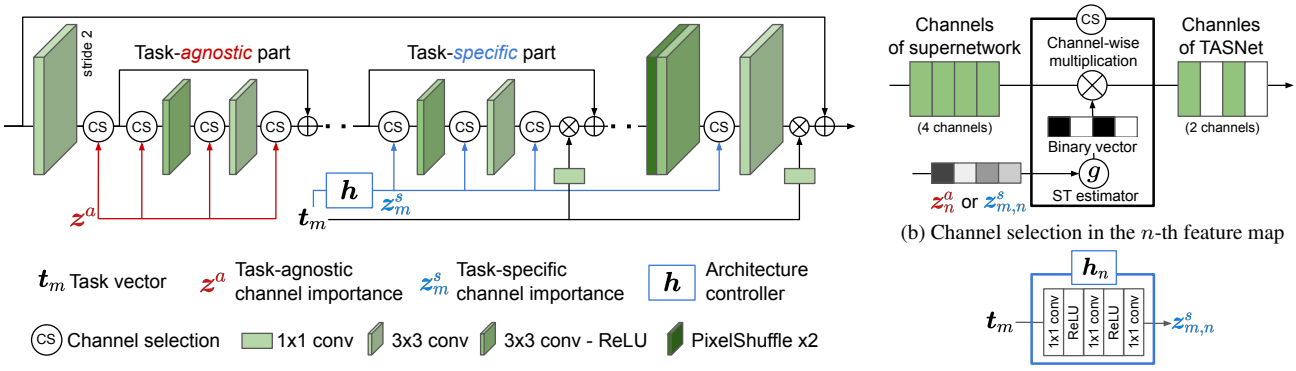
[43] Yu Zhang and Qiang Yang. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*, 2017. 3

Searching for Controllable Image Restoration Networks
– *Supplementary Document* –

A. Implementation details

Supernetwork architecture. We use the network architecture of CResMD as the supernetwork in the proposed search algorithm. Our supernetwork consists of 32 enhanced residual blocks which have a ReLU activation layer between two convolution layers with 64 filters of the kernel size 3×3. The first convolution layer with a stride of 2 downscales the input images, and the last upsampling module consists of PixelShuffle layer, two convolution layers, and a ReLU activation layer. Global skip connection adds the input image to the output of the upscaling module. A task vector scales the residual feature map in the location of 32 local connections and 1 global connection by a 1×1 convolution layer with channel-wise multiplication.

TASNet architecture. The proposed algorithm determines the number of shared layers and selects the channels of each shared or non-shared layer. Figure A(a) illustrates the TASNet architecture. For the shared layers (task-agnostic part), the channels that are not selected at the end of training are pruned in the final model. On the other hand, the non-shared layers (task-specific part) adaptively select their channels *w.r.t* the input task vector. During the training, the channels are virtually selected by channel-wise multiplication to the binary vectors, as described in Figure A(b). Our channel selection modules are located at all feature maps after the initial PixelShuffle layer of CResMD. The architecture controller consists of 3 fully-connected layers with ReLU activation function, as described in Figure A(c). In the task-agnostic part of the supernetwork, the residual scaling modules are removed to make the feature maps independent to specific tasks.



(a) Overview of TASNet architecture

(b) Channel selection in the n-th feature map

(c) Architecture controller in the n-th feature map

Figure A: TASNet architecture. (a) During searching for the number of shared early layers (task-*agnostic* part) in the supernetwork, z^a determines where to prune in the task-*agnostic* part. By contrast, z^s_m selects channels in the remaining layers (task-*specific* part) specialized in the m-th task-vector t_m (the control factor of restoration levels). We omit the notations for the feature map index n and the channel index c for simplicity. (b) In the network training phase, channel-wise multiplication between a binary vector and a feature map operates as virtual channel selection (CS) for the differentiable neural architecture search process. (c) Architecture controller consists of fully connected layers and predicts task-*specific* channel importance from the task vector.

Hyperparameters for the search algorithm. TASNet sets the hyperparameters α, γ, M, λ_1, and λ_2 as 0.9, 0.9, 64, $5\times e^{-11}$, and $1 \times e^{-2}$, respectively. The mini-batch consists of 64 image patches with 64×64 resolution. The initial learning rate is 1×10^{-4}. TASNet is trained for 1×10^6 iterations using Adam optimizer [45] with the learning rate decay of ×0.5 after the first half of training.

Image quality measure. In this work, we utilize three widely used image quality measures, PSNR, SSIM, NIQE [44], and BRISQUE [46] to evaluate the quality of images produced by models. PSNR and SSIM are full-reference measures in that the restored images are compared with the original clean images. On the other hand, NIQE and BRISQUE are no-reference evaluation metrics, in which the restored image quality is measured without referring to the original image. Images with higher

PSNR, higher SSIM, lower NIQE, and lower BRISQUE scores are considered to have better quality. However, measuring image quality during adjusting restoration levels has not been studied thoroughly. Thus, we visualize extensive qualitative results in both the main manuscript and the supplementary document.

Degradation in non-blind test set. For fair comparisons in non-blind setting, we construct CBSD68 dataset with the combinations of three levels and three types of degradation; Gaussian blur with $r \in \{0, 2, 4\}$, Gaussian noise with $\sigma \in \{0, 25, 50\}$, and JPEG compression with $q \in \{None, 60, 10\}$. Among the 27 combinations of degradation, we omit $(r, \sigma, q) = (0, 0, None)$ which generates identical images to the original. PSNR, SSIM, NIQE, and BRISQUE in all tables of this paper report the average scores on CBSD68 dataset with the 26 combinations of degradation.

Computation cost metric. We measure the computation costs of the networks in FLOPs and latency. FLOPs is a classical device-agnostic metric and exponentially increases by image resolution. Since latency is device-dependent, we measure latency on CPU with single-core (CPU latency (single)), CPU with multi-core (CPU latency (multi)), and GPU (GPU latency). We use Intel i7-5960X CPU which has 16 cores and GeForce RTX 2080 Ti GPU. The computation costs reported in this paper are average scores to generate images with 27 restoration levels unless otherwise mentioned. The task vectors $\boldsymbol{t} \in \mathbb{R}^3$ represent the 27 restoration levels by $t_d \in \{0, 0.5, 1\}$.

B. Additional experiments

Balancing the hyperparameters. Table A presents the ablation study of hyperparameters λ_1 and λ_2 which balance the trade-off between the network computation cost and the number of shared layers while minimizing Equation (8) of the main paper. The models trained with small λ_1 and large λ_2 have large portions of shared layers, and thus they are efficient in generating multiple (27) images (② *vs.* ③ and ⑤ *vs.* ④). In contrast, the models trained with the opposite balance between λ_1 and λ_2 are efficient for a single inference (② *vs.* ① and ⑤ *vs.* ⑥).

Table A: Ablation study of hyperparameters λ_1 and λ_2 on CBSD68.

Ex.#	λ_1	λ_2	#Shared layer	PSNR	FLOPs	
					Single inference	Multiple inferences
①		1×10^{-3}	18 %	25.67 dB	35.2 G	23.1 G
②	5×10^{-11}	1×10^{-2}	62 %	25.75 dB	52.9 G	7.5 G
③		1×10^{-1}	99 %	25.48 dB	125.5 G	4.8 G
④	5×10^{-12}		99 %	25.46 dB	154.6 G	6.0 G
⑤	5×10^{-11}	1×10^{-2}	62 %	25.75 dB	52.9 G	7.5 G
⑥	5×10^{-10}		16 %	25.50 dB	15.4 G	1.9 G

Extra qualitative results. We present more qualitative comparisons between CResMD and TASNet in the blind setting where users have to generate diverse restored images by controlling the restoration levels (task vectors) for unknown degradation of an input image. Recall that CResMD incurs three problems in this scenario: artifacts in the generated images, over-smoothed outputs, and uneven modulation across the task vectors. Figure B and C show that CResMD produces output images with undesired and visually unpleasing artifacts. Figure D presents less artifacts in the outputs of CResMD, but the outputs are over-smoothed compared to the outputs of TASNet even for the true task vector. Figure E also shows over-smoothed outputs for CResMD when restoring the input images with high restoration levels for denoising and dejpeg. By contrast, TASNet maintains the sharp textural details of the input image and removes visually unpleasing noise and compression artifacts of the input. Figure F exemplifies the problem of uneven modulation for CResMD. While CResMD produces images with negligible changes for lower values of deblurring level, it exhibits drastic changes for higher levels. In contrast to CResMD, TASNet demonstrates more even modulation across the different task vectors and generates smoothly-varying images. Figure G, H, and I presents modulation scenarios for a *real-word image* with unknown degradation, in which modulations with various task vectors are inevitable to find the visually pleasing images. These results demonstrate that CResMD sometimes generates severely destructive artifacts (especially in Figure G) and overly-smooth outputs (especially in Figure H) during the modulation process whereas TASNet generates plausible images for various task vectors.

Figure B: **Deblur modulation examples to the image with blur, noise, and jpeg compression**. Our TASNet generates diverse images with respect to the given restoration levels (task vectors). TASNet generates less auxiliary visual artifacts. The values of task vector denote restoration levels of (deblur, denoise, dejpeg), respectively.

Figure C: **Deblur and dejpeg modulation examples to the image with blur, noise, and jpeg compression**. Our TASNet generates diverse images with respect to the given restoration levels (task vectors). TASNet generates less auxiliary visual artifacts. The values of task vector denote restoration levels of (deblur, denoise, dejpeg), respectively.

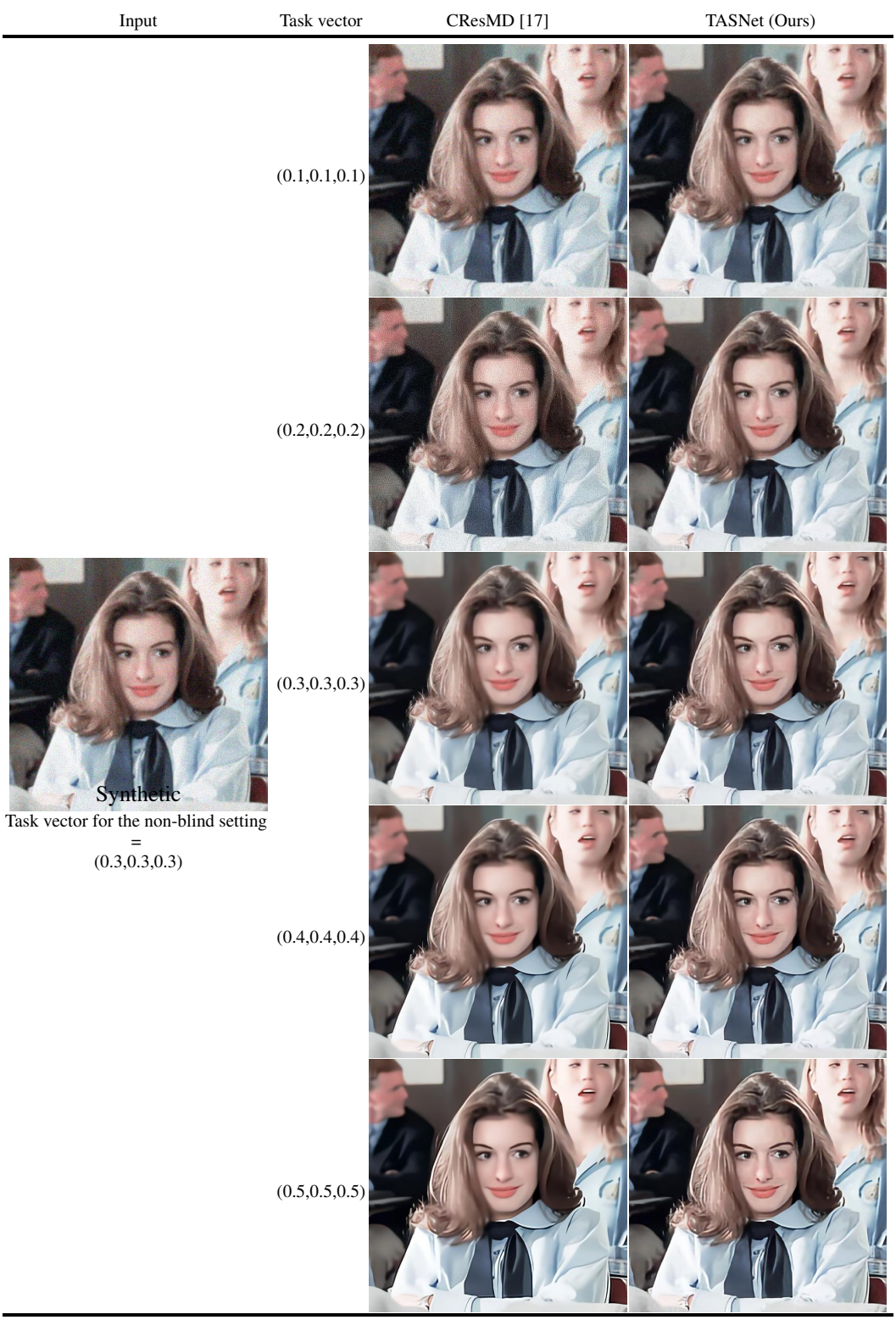


Figure D: **Deblur, denoise, and dejpeg modulation examples to the image with blur, noise, and jpeg compression**. Our TASNet generates diverse images with respect to the given restoration levels (task vectors). TASNet generates less auxiliary visual artifacts and over-smoothed textures. The values of task vector denote restoration levels of (deblur, denoise, dejpeg), respectively.

Figure E: **Denoise and dejpeg modulation examples to the image with noise and jpeg compression**. Our TASNet generates less over-smoothed textures. The values of task vector denote restoration levels of (deblur, denoise, dejpeg), respectively.

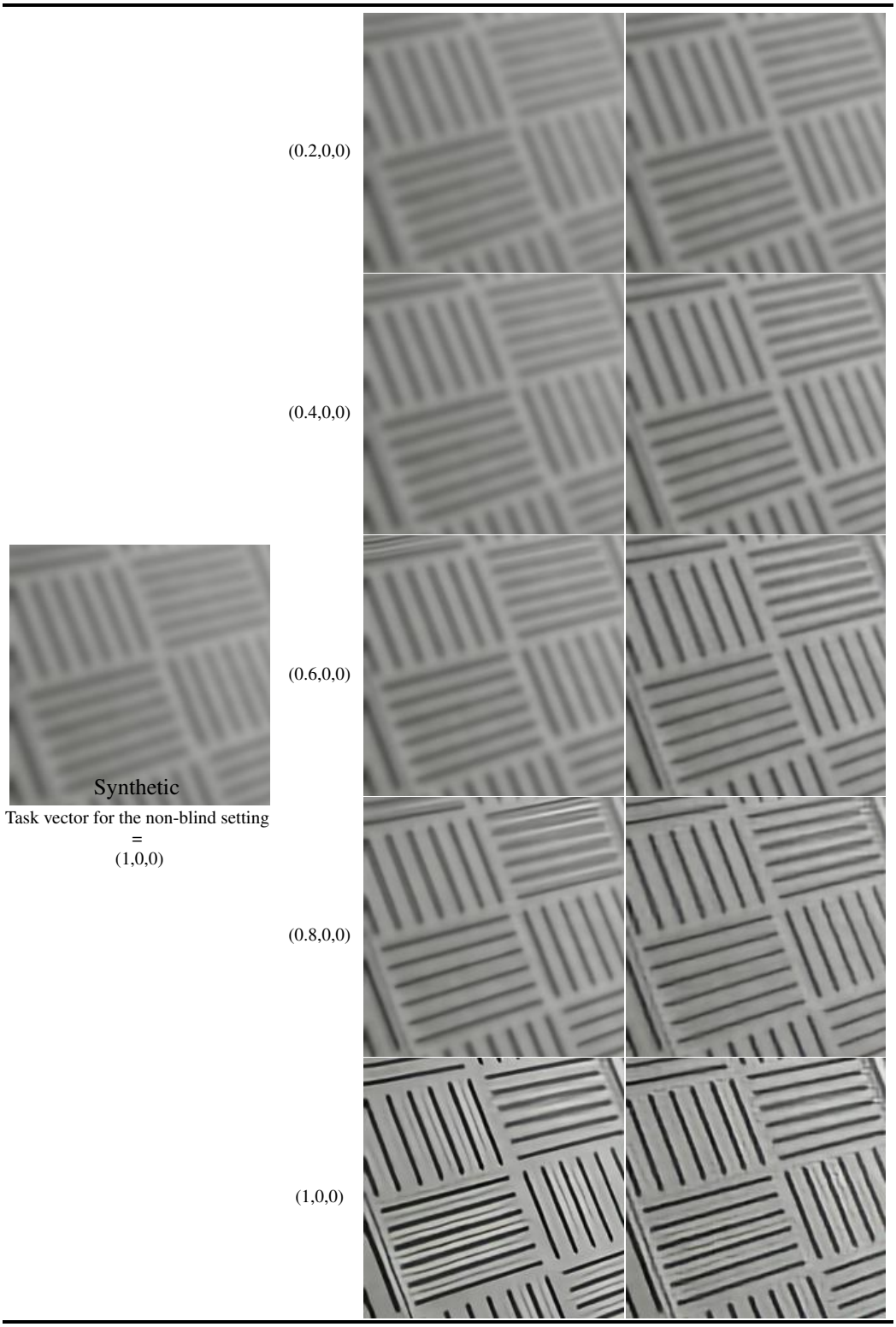


Figure F: **Deblur modulation examples to the image with blur**. Our TASNet generates evenly modulated images with respect to the given restoration level changes. The values of task vector denote restoration levels of (deblur, denoise, dejpeg), respectively.

Figure G: **Deblur modulation examples to the real world image on the Internet**. Our TASNet generates diverse images with respect to the given restoration levels (task vectors). TASNet generates less auxiliary visual artifacts. The values of task vector denote restoration levels of (deblur, denoise, dejpeg), respectively.

Figure H: **Denoise modulation examples to the real world image on the Internet**. Our TASNet generates diverse images with respect to the given restoration levels (task vectors). TASNet generates less over-smoothed textures. The values of task vector denote restoration levels of (deblur, denoise, dejpeg), respectively.

Figure I: **Deblur, denoise, and dejpeg modulation examples to the real world image on the Internet**. Our TASNet generates diverse images with respect to the given restoration levels (task vectors). TASNet generates less auxiliary visual artifacts and over-smoothed textures. The values of task vector denote restoration levels of (deblur, denoise, dejpeg), respectively.

References

[44] Alan C. Bovik Anish Mittal, Rajiv Soundararajan. Making a completely blind image quality analyzer. *SPL*, 2013. 1

[45] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 1

[46] Anish Mittal, Anush K. Moorthy, and Alan C. Bovik. Blind/referenceless image spatial quality evaluator. In *ASILOMAR*, 2011. 1