Pyramid Architecture Search for Real-Time Image Deblurring

Xiaobin Hu[1,2,7], Wenqi Ren[2]*, Kaicheng Yu[3,6], Kaihao Zhang[4], Xiaochun Cao[2], Wei Liu[5], Bjoern Menze [1,7]
[1]Informatics, TU München [2]SKLOIS, IIE, CAS [3]CVLab, EPFL [4]Australian National University
[5]Tencent Data Platform [6]Abacus.AI [7]Quantitative Biomedicine, University of Zurich

Abstract

Multi-scale and multi-patch deep models have been shown effective in removing blurs of dynamic scenes. However, these methods still suffer from one major obstacle: manually designing a lightweight and high-efficiency network is challenging and time-consuming. To tackle this obstacle, we propose a novel deblurring method, dubbed PyNAS (pyramid neural architecture search network), towards automatically designing hyper-parameters including the scales, patches, and standard cell operators. The proposed PyNAS adopts gradient-based search strategies and innovatively searches the hierarchy patch and scale scheme not limited to cell searching. Specifically, we introduce a hierarchical search strategy tailored to the multi-scale and multi-patch deblurring task. The strategy follows the principle that the first distinguishes between the top-level (pyramid-scales and pyramid-patches) and bottom-level variables (cell operators) and then searches multi-scale variables using the top-to-bottom principle. During the search stage, PyNAS employs an early stopping strategy to avoid the collapse and computational issues. Furthermore, we use a path-level binarization mechanism for multi-scale cell searching to save the memory consumption. Our primary contribution is a real-time deblurring algorithm (around 58 fps) for 720p images while achieves state-of-the-art deblurring performance on the GoPro and Video Deblurring datasets.

1. Introduction

Blind motion deblurring is an ill-posed problem aiming to rehabilitate the sharp image from a degraded image caused by depth variation, camera shakes, and object motions [21, 26, 35]. In the past decades, image deblurring has attracted massive attention from the computer vision community, especially in applications of surveillance, remote sensing, and cameras. Traditional deblurring approaches remove the undesirable blur via the blur kernel estimation

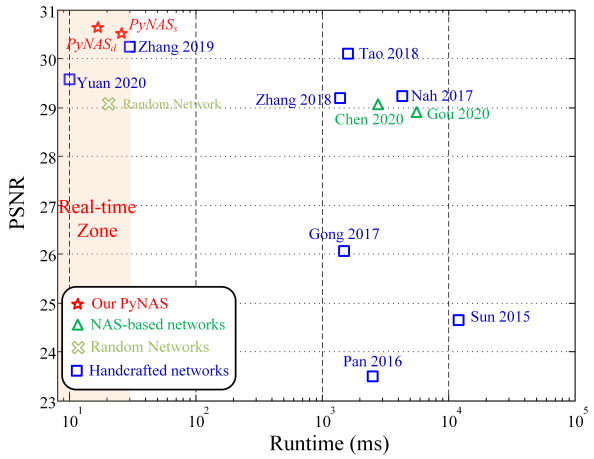


Figure 1. PSNR vs. test runtime of state-of-the-art deep motion deblurring methods and the proposed PyNAS on the GoPro dataset [21]. The pink zone indicates that the inference time of a 720p image is less than 33ms (i.e., 30fps). The architecture found by our PyNAS achieves better performance and less inference time (17 ms) than other handcrafted or NAS-based networks.

[7, 25], which approximates the hidden degradation knowledge. However, obtaining a satisfactory blur kernel ramains as an open problem.

Deep learning-based models [31, 35, 8] have verified the superiority in learning the regression between blurry inputs and the corresponding sharp images. Especially, the 'coarse-to-fine' scheme has been applied to deblurring scenarios via a multi-scale architecture to exploit the deblurring priors at different levels. However, there still exist some challenges: i) expensive runtime caused by the large number of training parameters due to large filter size; ii) structure redundancy attributed to the ineffectual increasing depth for very low-resolution or small patch input in a multi-scale model [21].

Recently, a growing interest is witnessed in developing algorithms to automatically design deep learning architectures instead of the manual process. Architectures found by the searching algorithms have achieved highly competitive performance especially on high-level tasks such as image classification [16], object detection [37, 6] and seman-

*Corresponding author

tic segmentation [41, 17]. Inspired by this, we aim to figure out an effective algorithm to tackle the above deblurring challenges. Specifically, we design a searching algorithm to automatically optimize the lightweight pyramid structure hyper-parameters, including the scale-pyramid and patch-pyramid. Our main contributions are summarized as:

- We propose a lightweight multi-scale pyramid neural architecture search approach for image deblurring, termed PyNAS. To our knowledge, this is the first attempt to design a multi-scale architecture search algorithm to handle dynamic scene deblurring tasks.
- The proposed PyNAS innovatively includes the pyramid structure (scale-pyramid and patch-pyramid schemes) into the search space to tackle the challenge of non-uniform motion blurs in dynamic scenes.
- We define the hierarchical relationship among optimization variables as the top-level (patch and scale scheme) and bottom level (basic operators), and we train the model with a top-to-bottom approach.

We apply our algorithms on the GoPro and VideoDeblurring datasets for evaluation. We qualitatively and quantitatively evaluate the proposed algorithm on the real-time video deblurring task. Shown in Figure 1, the proposed network achieves better performance compared to SOTA methods using less inference time.

1.1. Related Work

Because of the spatially invariant blur kernels, conventional image deblurring methods fail to remove non-uniform motion blurs [13, 12]. Besides, the long-time processing inference cannot meet the real-time demand of video deblurring. CNNs have been used in non-uniform image deblurring to deal with the motion blur in a time-efficient inference [24, 29, 39, 47, 5]. Considering the difference in network architectures and patch schemes, we categorize these methods as single-scale, multi-scale, and multi-patch algorithms.

Single-Scale Networks. The single-scale deblurring methods [31, 34, 48, 30] aim to recover highly-realistic images mainly based on well-developed network blocks for high-level vision tasks, which usually leading to a heavy and time-consuming network. For example, following residual learning [10], DeblurGAN [14] and DeblurGAN-V2 [15] remove blurs by adversarial learning. The attention mechanism is also introduced into image restoration [49]. To obtain the multi-scale priors for the singe-scale architecture, some researchers refer to the U-Net [28] and dilated convolution [40] for image restoration tasks [15].

Multi-Scale Networks. Multi-scale architectures have been verified effective in image restoration [8, 35], especially in image deblurring [21]. It can restore sharp images in a progressive manner by a network at each scale. Essentially, multi-scale architecture is to mimic the conventional coarse-to-fine framework to decompose a challenging task into smaller easier subtasks. For example, Nah *et al*. [21] proposed a multi-scale CNN for image deblurring by imitating the coarse-to-fine strategy in conventional deblurring approaches. A scale-recurrent network (SRN-DeblurNet) is proposed in [35] by exploiting ConvLSTM to aggregate feature maps from coarse to fine scales. Gao *et al*. [5] proposed an effective selective sharing scheme for constraining the deblurring network and a nested skip connection structure for the nonlinear transformation. Zamir *et al*. [43] introduced a multi-stage architecture progressively learning restoration functions for degraded images.

Although the multi-scale network can boost the performance of image deblurring, all the above existing algorithms are designed in a labor-intensive handcrafted manner. It leaves an open and challenging question how to optimize a multi-scale architecture in the aspects of removing ineffectual scale depth and unnecessary large filters to reduce the runtime while keeping competitive performance.

Multi-Patch Networks. Zhang *et al*. [45] proposed a hierarchical multi-patch scheme (DMPHN) to keep spatial information without any image down-sampling like [21]. The multi-patch input for each scale is generated by dividing the original blurry input into multiple non-overlapping patches, shown in Fig. 2. Following this work, some recent researches [43, 33] also utilized the multi-patch scheme for multi-scale image deblurring. Intuitively, for non-uniform deblurring, the blur kernel of global image is different from the blur kernel of local patches. Compared with the information loss of blur kernel due to the image degradation of multi-scale framework, we utilize the multi-patch scheme to exploit the blur kernel priors and better represent the global non-uniform knowledge. However, all the above work fixed patch scheme in a manual manner and the further optimization of the multi-patch mechanism is still unexplored so far.

Neural Architecture Search (NAS). Neural architecture search aims to automatically design a high-performance architecture such that it largely eliminates the tedious and heuristic manual design of neural architecture. Early work introduced evolutionary algorithms (EAs) to optimize neural architecture by iteratively mutating a population of candidate architectures [18]. Reinforcement learning (e.g., policy gradients[51, 37, 2, 38] and Q-learning [50]) is an alternative algorithm to optimize potential architectures by exploring the search space. But these methods require massive computation in the procedure of searching. Thus, speed-up techniques [11] (e.g., hyper-networks [44] and shared weights [27]) were proposed to redress this issue.

Inspired by this idea, we propose a pyramid neural architecture search network to automatically design a multi-scale and multi-patch mechanism. Closest to ours is the work of

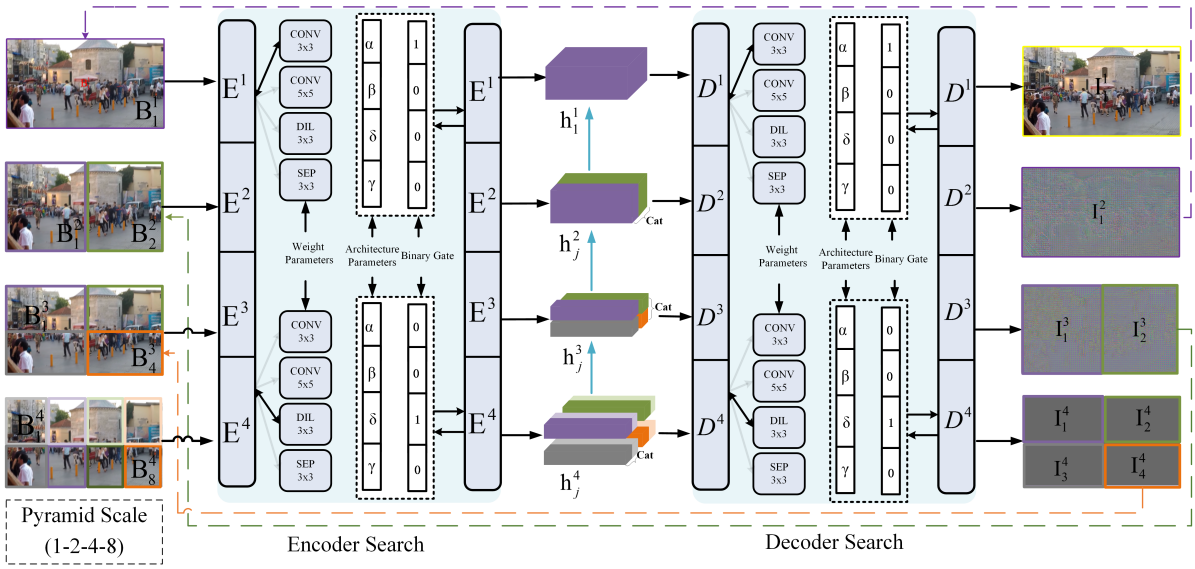


Figure 2. The proposed pyramid architecture search (PyNAS) algorithm. We use the pyramid patch scheme (1-2-4-8) and scale depth (4) for the illustration. The non-overlap multi-patch hierarchy is fed to the network. PyNAS searches the whole encoder E_i and decoder D_i structure of each scale from the operator candidates. $\mathbf{I}^i$, $\mathbf{B}^i$, $\mathbf{h}^i$ are the estimated latent sharp, blurry images, and hidden features at the i-th scale. Note that our PyNAS finds a better pyramid network architecture (1-3-9) using less inference time and shallower scale depth.

ProxylessNAS [1] and DARTS [19], in which the authors exploited the gradient descent strategy after a continuous relaxation on the architecture representations and achieved competitive performance on the image classification task. Our method differs from [1] and [19] in several important ways: i) Considering the significant advantage of the specialized multi-scale [21] and multi-patch [45] mechanism for image deblurring, we innovatively extend the search space to include the multi-scale structure and patch composition, and then search a high-performance hierarchical scale-pyramid and patch-pyramid network for image deblurring; ii) our proposed multi-scale search strategy is different from the existing NAS search strategies including the CLEARER [8] and HiNAS [46]. Specifically, we define the hierarchical relationship among optimization variables as the top-level (patch and scale schemes) and bottom level (basic operators) when some work [1, 19] assumed that all variables are parallel; iii) we train the model in a top-to-bottom manner and adopt the random policy for the top-level variables initialization and the gradients-based policy for the bottom-level variables searching, respectively. Besides, shown in algorithm 1, we run the iterative optimization and stack previous well-searched networks until all scale networks are searched.

2. Proposed Method

2.1. Problem Formulation

Blur artifacts are caused by different reasons (e.g., camera shakes and object movement). Typically, the mathematical formulation of the blurring process can be described as:

$$\mathbf{B} = \mathbf{KS} + \mathbf{n}, \tag{1}$$

where $\mathbf{B}$, $\mathbf{S}$ and $\mathbf{n}$ denote blurry, sharp images, and additive noise, respectively. $\mathbf{K}$ is the blur kernel. We adopt a multi-patch structure across multiple scales in the coarse-to-fine strategy to solve the deburring formulation. Specifically, at each scale, sharp latent images are generated by this scale network, and then imported and concatenated as the input of the next scale sub-network, which can be expressed as:

$$\mathbf{I}^i, \mathbf{h}^i = Net_i(\mathbf{B}^i, \mathbf{I}^{i+1}, \mathbf{h}^{i+1}; \theta_i), \tag{2}$$

where i is scale index and the first scale $i = 1$ is the finest scale. As shown in Figure 2, $\mathbf{I}^i$ and $\mathbf{B}^i$ are the estimated latent sharp and blurry images at the i-th scale, respectively, while $\mathbf{h}^i$ means hidden state features across scales. Net_i is the i-th scale network architecture consisting of encoder E^i and decoder D^i with training parameters θ_i.

Following [43, 33, 45], we introduce the multi-patch hierarchy as input to keep the same spatial resolution even at different scales, which alleviates expensive runtime on the deconvolution/upsampling operation. Further, the main idea that setting the number of patches at each scale different is to make the coarse scales focus on local information to produce residual information for the next finer scale. We assume that the scale depth of the network is N and accordingly its multi-patch hierarchy scheme is $(1, ..., P_i, ..., P_N)$. Patch scheme P_i denotes the number of image non-overlapping patches at the specific scale i. It is worth noting that our algorithm can automatically optimize the scale depth and multi-patch hierarchy scheme.

The overall architecture of our PyNAS network is illustrated in Figure 2. The deblurring operation starts at the bottom scale N=4. In this scale, the whole blurred image B^1 is divided into P^N non-overlapping patches B_j^N ($j = 1, ..., P_N$). Each scale of our network consists of one encoder E and one decoder $\mathcal{D}$. Then, we feed the patches into the N-th scale encoder E^N to produce the latent feature representation:

$$\mathbf{h}_j^N = \mathbf{E}^N(\mathbf{B}_j^N), \quad j \in 1, .., P_N, \tag{3}$$

Afterward, we concatenate (Num=P_N/P_{N-1}) adjacent features to form a new feature representation $\widehat{\mathbf{h}}_k^N$ as follows:

$$\widehat{\mathbf{h}}_k^N = \mathbf{h}_j^N \oplus ... \oplus \mathbf{h}_{j+\text{Num}}^N, \tag{4}$$

where $k \in 1, .., P_{N-1}$ and $\oplus$ means the concatenation operator. Then, $\widehat{\mathbf{h}}_k^N$ is passed through decoder $\mathcal{D}_N$ to produce $\mathbf{I}_k^N = \mathcal{D}_N(\widehat{\mathbf{h}}_k^N)$ which has the same spatial size as the patches $\mathbf{B}_k^{N-1}$ (e.g., $\mathbf{I}_4^4$ and $\mathbf{B}_4^3$ in Figure 2). The features at all scales are concatenated along the spatial dimension. In other words, fitting together neighboring patches forms a larger patch.

2.2. Architectural Search Methodology

To effectively process hierarchical patches from each scale, we utilize a gradient-based NAS technique to optimize each scale structure, scale depth, and multi-patch hierarchy scheme. In this section, we first introduce how to search for architecture cells using a continuous relaxation and path binarization. Then we explain how to define the scale depth and multi-patch hierarchy scheme space. Lastly, we elaborate on our multi-scale search strategy and loss function.

2.3. Cell Architecture Search

Continuous Relaxation. Following the continuous relaxation in [1], in each block integrated all possible layer types, the output tensor T_i^l is linked to all the input tensor I_i^l through a searched operation $O_{j\to i}$

$$T_i^l = \sum_{T_j^l \in I_i^l} O_{j\to i}(T_j^l), \tag{5}$$

In order to make the search space continuous, we estimate the searching step $O_{j\to i}$ with a continuous relaxation, yielding $\overline{O}_{j\to i}$:

$$\overline{O}_{j\to i} = \sum_{O\in\mathcal{O}} \alpha_{j\to i} O_{j\to i}(T_j^l), \tag{6}$$

where $\sum_{O\in\mathcal{O}} \alpha_{j\to i}$=1 and $\mathcal{O} \in O^1, O^2 ..., O^S$ are all possible S layer candidates, $\alpha_{j\to i}^k$ denotes the weight of operator O^k.

Path Binarization. In case of the memory limitation of hardware with a large design space, we employ the operation based on path binarization. Specifically, N real-valued architecture parameters α_i denote the path weights and are then transformed into binary gates:

$$g = \text{binarize}(p_1, .., p_N) \begin{cases} [1, 0, ..., 0] \text{ with probability } p_1, \\ ... \\ [0, 0, ..., 1] \text{ with probability } p_N, \end{cases} \tag{7}$$

Applying the binary gates g, the output of the mixed operation is reformulated as :

$$m_{\mathcal{O}}^{\text{Binary}} = \sum_{i=1}^{N} g_i o_i(x) = \begin{cases} o_1(x) \text{ with probability } p_1, \\ ... \\ o_N(x) \text{ with probability } p_N, \end{cases} \tag{8}$$

Shown in Eqs. 7 and 8, instead of the path weights, by using the binary gates, only one path of activation is active in memory at run-time and the memory requirement is reduced to an acceptable level.

Hierarchical Multi-Scale Search Space. Considering the contradiction between real-time deblurring and large multi-scale computation burden, we rely on computation-efficient operation with an adaptive receptive field such as dilated convolutions and without using large kernel convolutions. Besides, for each scale, the input patch size is different and, accordingly, the demand of receptive field should be optimized. In this paper, we pre-define the following 5 types of basic operators and two multi-scale optimization variables:

- Pyramid variable: scale depth $1 \leq D \leq 4$;
- Pyramid variable: multi-patch hierarchy scheme $[1, p_2, .., p_N]$, p_N denotes the patch number at N scale;
- Conv operators: 3 $\times$ 3 convolution;
- Conv operators: 5 $\times$ 5 convolution;
- Dilation operators: 3 $\times$ 3 convolution with dilation rate of 2;
- Separable operators: 3 $\times$ 3 separable convolution;
- Zero operators: no connection and return zero.

Each convolution operation is followed by a ReLU activation layer. Since batch normalization tends to corrupt the pixel-level correlation, we abandon the batch normalization and employ the mini-batch for training. However, the training stage using mini-batch leads to great computation time and largely increases the optimization difficulty of the multi-scale structure.

2.4. Multi-Scale Search Strategy

Based on the characteristics of the multi-scale deblurring network, we propose a hierarchical search strategy, separating all optimization variables into bottom-level (basic operators) and top-level parameters (scale-pyramid and patch-pyramid). First, we randomly initialize the top-level parameters (scale and patch numbers). We note that overly small

Algorithm 1 Multi-Scale Search Strategy for our system.

Input:

The multi-scale pyramid search space M including the scale depth (n), patches scheme (p), and the single-scale layer search space S.

Output:

The well-optimized multi-scale architecture M_{final} under architecture constraints;

while iteration $\leq$ I **do**
  P_0: Initialize the multi-scale scheme M_0 including the scale number (n) and patches scheme (p);
  for $i = 1; i < n; i++$ **do**
    S_0: Initialize the i-th scale network structure;
    Update: Gradient-based $MSE_{loss} = S_i()$ network search;
    S_i: i-th scale optimized candidate network;
    M: Stack the single-scale candidates S_i;
  end for
end while
return M_{final}.

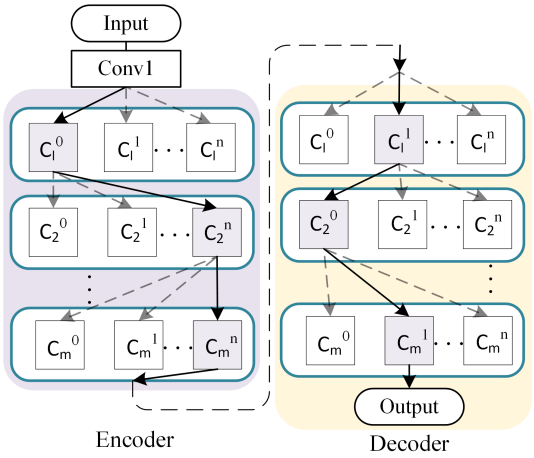


Figure 3. Encoder (left) and decoder (right) structure searching from the search space. For real-time image deblurring, our PyNAS aims to obtain a compact network consisting of compact cells and only keeping one cell in each layer.

patches (16×16) are not beneficial for removing the motion bur. Therefore, the maximum patch number of the hierarchy scheme is set as 16 to ensure that the patch size of the last scale is larger than 16×16.

After determining the pyramid-scales and pyramid-patches scheme, we first optimize the coarsest scale network and then stack the well-optimized coarse scales to further optimize the next finer scale. As illustrated in Figure 3, each scale network consists of encoder and decoder network searching and only keeps one cell in each layer for obtaining a compact architecture. We run the iterative optimization until all scale networks are searched. Then we train the well-optimized network with the initial top-level parameters empirically around 210 epochs to calculate MSE loss. Afterward, we update the scale depth and patch numbers to search better top-level parameters, shown in Algorithm 1. To speed up the search procedure and avoid the collapse when the number of search epochs becomes large, we employ the early stop principle during optimizing. For top-level variables, we build up a judgement module to ensure that the same schemes are not repeatably generated.

Table 1. Quantitative results on the GoPro test dataset. Size and Runtime are addressed in MB and Millisecond (ms). The reported time is only network inference time excluding the time of writing generated images to disk. We expand the search space of the hierarchy multi-patch scheme to odd types (such as [1-3-9]). The best results are highlighted in bold and the second best is in underline.

GoPro dataset				
Models	PSNR	SSIM	Size	Runtime
Sun *et al.* [34]	24.64	0.8429	54.1	12000
Nah *et al.* [21]	29.23	0.9162	303.6	4300
Zhang *et al.* [47]	29.19	0.9306	37.1	1400
Tao *et al.* [35]	30.10	0.9323	33.6	1600
Zhang *et al.* [45]	30.25	0.9351	29.0	30
Yuan *et al.* [42]	29.57	0.9338	**3.1**	**10**
Ours (PyNAS$_s$)	<u>30.51</u>	<u>0.9391</u>	<u>20.7</u>	26
Ours (PyNAS$_d$)	**30.62**	**0.9405**	35.9	<u>17</u>

2.5. Multi-scale Loss Function

Our multi-scale loss function is based on the Mean Square Error (MSE) and then modified by considering the search variable and multi-scale architecture.

$$\mathcal{L} = \sum \|Net\{N, P, \theta, C\} - \mathbf{G}\|^2, \tag{9}$$

where N, P, θ, C mean the scale depth, hierarchy patch scheme, network weights, and cell operators, respectively. We follow the principle of residual learning and regard the intermediate outputs as the residual information capturing the image priors at different scales. Consequently, we calculate our loss only at the finest scale.

3. Experiments

3.1. Datasets

GoPro Dataset [21] is used to verify the deblurring performance of our algorithm. It consists of 3214 pairs of blurred and sharp images extracted from 33 sequences captured at 720 ×1280 resolution. The blurred images are generated by averaging successive latent frames to produce varied blur. We follow the same strategy in [21], using the 2103 image pairs for training and the remaining 1111 pairs for test.

VideoDeblurring Dataset [31] contains videos captured by different devices such as GoPro Hero 4, iPhone 6s, and

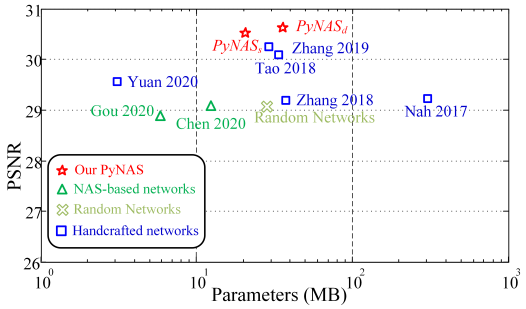


Figure 4. The Peak signal-to-noise ratio (PSNR) vs. parameters of state-of-the-art networks and our PyNAS on the GoPro dataset.

Nexus 5x. The quantitative subset consists of 6708 blurry frames and corresponding sharp images from 71 videos. Following previous work [31], total videos are split into 61 training and 10 testing videos.

3.2. Experimental Setting

Searching Settings. We perform network search on the GoPro dataset and randomly choose 10% of training samples as the validation set. We train the warm-up stage to update weights of convolutional layers and then the second stage is used to optimize parameters of the neural architecture. At second stage, we search for basic operators (bottom variables) at most 80 epochs with batch size of 32 and apply early stoping to avoid the model collapse. We employ standard SGD optimizer and set the learning rate as 0.0005 with the cosine annealing strategy [20] in convolutional kernels updating. For the architecture update, Adam optimizer is used and the learning rate is set to 0.001.

Training Settings. All the training experiments are implemented in Pytorch with a single TITAN XP GPU. For random crop, the patches of 256×256 are randomly cropped from input images but 288×288 for the odd patch scheme (for example, $[1, 3, 9]$). The initial learning rate is 0.0001 and the decay rate is 0.1 with the Adam solver. We also normalize images to the range of [0,1] and subtract 0.5 to keep the pixel-level knowledge. We train the model for 3000 epochs with the mini-batch size 6.

3.3. Comparisons with State-of-the-arts

We compare our networks obtained by PyNAS with other recent state-of-the-art deblurring methods: a convolutional neural network for nonuniform motion blur removal [34], a deep multi-scale convolutional neural network [21], spatially variant recurrent neural networks [47], a scale-recurrent network [35], a deep stacked hierarchical multi-patch network [45], and a spatially variant deconvolution network [42]. We propose two models: one is the lightweight (PyNAS$_s$) which searches the identical cells for each scale and can share the weight among scales. The other is PyNAS$_d$ which searches the individual cell structure of each scale and then stacks them together. The learned PyNAS$_s$ is a pyramid-shape (1-2-4-8) network and mainly consists of dilation blocks and standard convolution blocks, while PyNAS$_d$ is a 1-3-9 patch scheme network and contains massive dilation blocks and convolution blocks with large kernel size. Detailed configurations (e.g., kernel size and operators) of PyNAS$_d$ and PyNAS$_s$ can be found in the supplementary material.

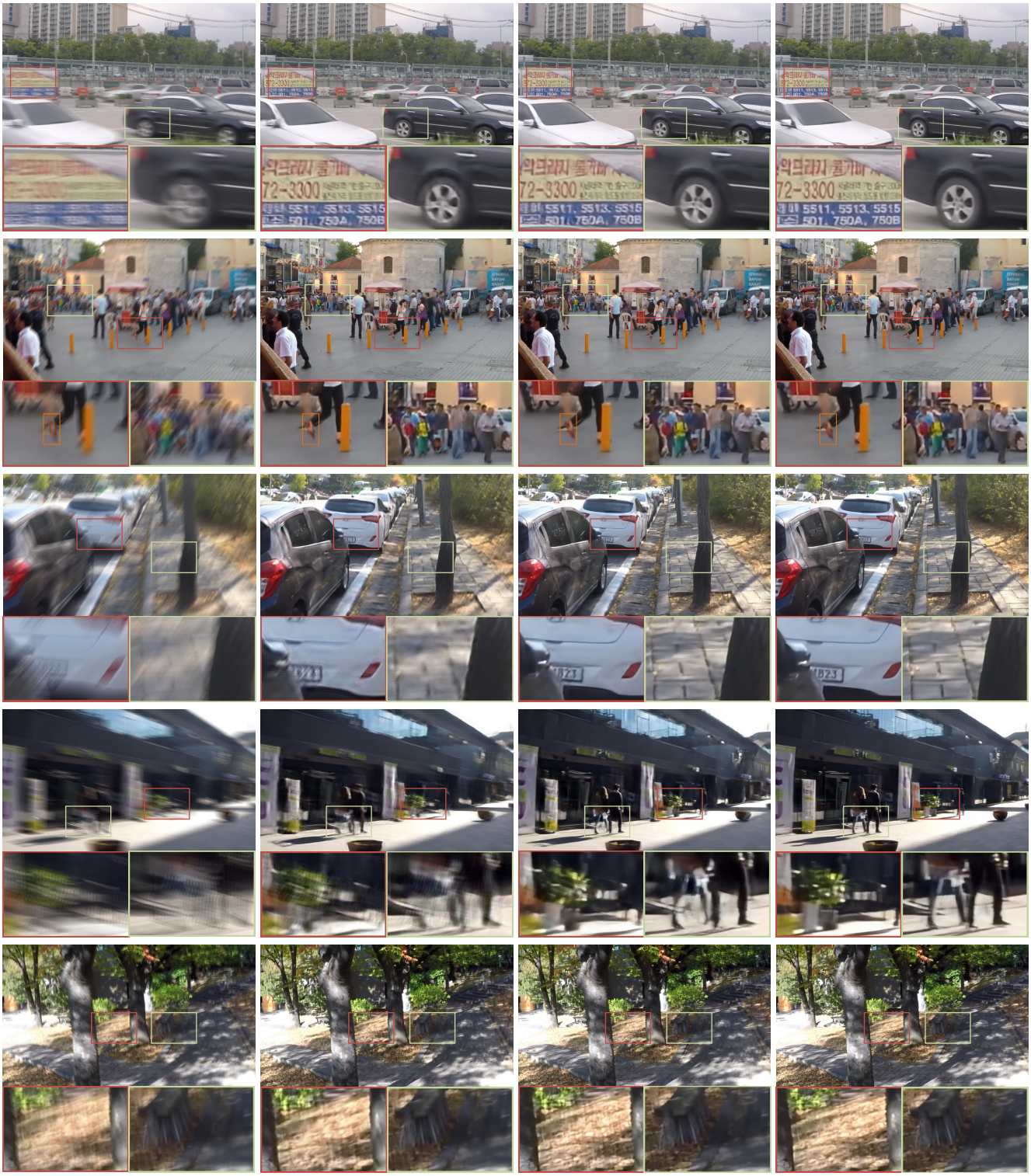
Quantitative Results. As shown in Table 1, our PyNAS$_d$ (different structures of each scale) achieves significantly better results (1.1 dB higher than the most recent real-time deblurring method ([42]) while maintaining the second fastest runtime $17ms$ for a 720p image compared with [42]. It's worth noting that both our PyNAS multi-scale architectures (PyNAS$_d$ and PyNAS$_s$) obtained by neural architecture search perform much better than the previous manual multi-scale architecture ([45] and [21]) and meanwhile decreasing the inference time. Also shown in Table 2, the quantitative results on the VideoDeblurring dataset indicate that our PyNAS methods achieve the better generalization performance than the very recent method ([45]).

Qualitative Evaluation. The five deblurred examples of the GoPro dataset [21] are shown in Figure 5. We show the visualization of different models for images containing large motion blur and zoom in the main object. Compared with recent deep learning models, the rehabilitated images of our method are clearer and sharper at the edges. The contents of our deblurred images are well recovered, e.g., the numbers of advertisement and license plate are deblurred properly, while the others fail to show clear numbers.

Real-Time Deblurring. Runtime is expressed as *ms* and only calculated by the CNN runtime (writing generated images to disk is not considered). In order to make the persistence of vision and produce the illusion of moving images, it requires a short time to process a blurred image (nowadays, usually 33ms for 30fps or 41ms for 24fps). For the main motivation of real-time video deblurring, only our models, [42] and [45] meet the real-time requirement to process the $720p$ images. Moreover, our PyNAS$_d$ can deblur a 1280×720 image with nearly **0.017s** per image, which is 40× than Tao *et al*. [35]. Our PyNAS models achieve the real-time deblurring and represent high performance of motion deblurring (over 30.50).

3.4. Multi-Scale Search Strategy vs. Random Policy

To evaluate the multi-scale searching strategy of our PyNAS compared with random policy, we randomly sample 10 models from all search spaces. From Table 3, our PyNAS can search a well-optimized model (1.5dB higher) than the model obtained by random policy. It verifies the effectiveness of multi-scale search strategy for multi-scale network architecture.



(a) Input (b) J. Zhang *et al*. [47] (c) H. Zhang *et al*. [45] (d) Ours (PyNAS$_d$)

Figure 5. Visual comparison with state-of-the-art deblurring methods from the GoPro dataset. The first column are the blurry images, the second column are the deblurred images of [47], and the third column are results of [45]. The last column shows results generated by our PyNAS$_d$, which achieves the best performance compared with the others.

Table 2. Quantitative results (PSNR) on the VideoDeblurring dataset. Our models are trained on the GoPro dataset and then generalized on the VideoDeblurring dataset.

VideoDeblurring Dataset											
Models	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	Average
Input	24.14	30.52	28.38	27.31	22.60	29.31	27.74	23.86	30.59	26.98	27.14
PSDeblur	24.42	28.77	25.15	27.77	22.02	25.74	26.11	19.75	26.48	24.62	25.08
WFA [4]	25.89	32.33	28.97	28.36	23.99	31.09	28.58	24.78	31.30	28.20	28.35
Su *et al*. [31]	25.75	31.15	29.30	28.38	23.63	30.70	29.23	25.62	31.92	28.06	28.37
Zhang *et al*. [45]	29.89	33.35	31.82	31.32	26.35	32.49	30.51	27.11	34.77	30.02	30.76
Ours (PyNAS$_d$)	**30.11**	**33.52**	**31.92**	**31.54**	**26.44**	**32.73**	**30.69**	**27.51**	**35.07**	**30.45**	**31.01**

Table 3. The quantitative analysis: the superiority of our multi-scale search strategy compared with ten random CNN models and other low-level NAS algorithms.

GoPro Dataset		
Models	PSNR	SSIM
Average of random models	29.11	0.9251
NAS-DIP [3]	29.01	0.9176
CLEARER [8]	28.96	0.9188
PyNAS$_s$	30.51	0.9391
PyNAS$_d$	**30.62**	**0.9405**

3.5. Comparisons with Other Low-level NAS

Considering that our work is the first attempt for dynamic scene deblurring tasks, we compare our PyNAS with other low-level NAS methods [23, 22, 46, 36, 32], such as super-resolution [9], denosing[8], and dehazing [3] tasks. Due to its compact module search space, CLEAR takes eight GPU hours to find a satisfactory network. NAS-DIP takes about 3 days (72 GPU hours) to find the best architecture. Compared with CLEAR and NAS-DIP, our PyNAS$_s$, which searches the identical cells for each scale, only takes 8 GPU hours in searching while our PyNAS$_d$ searching the individual cell structure of each scale takes about 36 GPU hours. As shown in Table 3, our PyNAS has obtained much better results (around 1.5dB) than other low-level NAS. This mainly attributes to the specialized pyramid-patches and pyramid-scale searching mechanism designed for dynamic scenes suffers from spatially-invariant blurs [45]. We employ the multi-scale and multi-patch scheme to decompose a challenging task into easier subtasks via dividing the globally spatially-invariant kernel into locally patch-based spatially-invariant kernels.

3.6. Architecture Analysis

In this subsection, we analyze the main characteristics of architectures designed by our PyNAS. We can find that:

- In the multi-scale network found by our PyNAS, the first and second scales are inclined to have a similar structure combined by the large kernel and dilation convolution, while the third and fourth scales are prone to having a similar structure consisting of the smaller kernels and dilation convolution. This phenomenon is mainly caused by the input patch size of each scale. Accordingly, the various requirements of receptive field are needed in each scale network.
- The following aspects lead to our PyNAS fast runtime: i) searching a suitable pyramid parameters with shallow scale depth (d=3) and patch scheme (p=1-3-9) and selecting small-size convolutional filters; ii) cutting off unnecessary links, e.g., skip or recurrent connections; iii) reduced the number of upsampling/deconvolution between convolutional features.
- To decrease the massive search time and computation of each scale, we define the PyNAS$_s$ search model assuming that each scale inherits the same network architecture. Compared with the model PyNAS$_d$ that has different network architectures for each scale, PyNAS$_s$ still gets satisfactory results for real-time video deblurring but only occupies around 1/4 search time of PyNAS$_d$.

4. Conclusions

In this paper, we proposed a pyramid neural architecture search algorithm (PyNAS) to optimize hyper-parameters (hierarchy patch, scale depth, and cells scheme) of the multi-scale network for the real-time deblurring task. The multi-scale and multi-patch mechanism are specialized design to tackle the challenge of non-uniform motion blur for dynamic deblurring tasks. A multi-scale search strategy is exploited to sequentially search the multi-parameters from the top (scale depth and patch scheme) to bottom variables (cells operator). To make PyNAS both memory and computation efficient, the path binarization and early stoping strategies are used. Our architecture obtained by the proposed PyNAS achieves better performance compared with the state-of-the-art algorithms with faster inference time. Further, our algorithm is applicable in real-time deblurring, only 17ms run time for one 720p image (i.e., 58fps).

Acknowledgments. This work was supported by the National Key RD Program of China under Grant 2020YFB1406704, National Natural Science Foundation of China (No. 61802403, 62025604, U1936210, 61971016).

References

[1] Han Cai, Ligeng Zhu, and Song Han. Proxylessnas: Direct neural architecture search on target task and hardware. In *International Conference on Learning Representations*, 2018. 3, 4

[2] Yukang Chen, Tong Yang, Xiangyu Zhang, Gaofeng Meng, Xinyu Xiao, and Jian Sun. Detnas: Backbone search for object detection. *arXiv preprint arXiv:1903.10979*, 2019. 2

[3] Yun-Chun Chen, Chen Gao, Esther Robb, and Jia-Bin Huang. Nas-dip: Learning deep image prior with neural architecture search. In *European Conference on Computer Vision (ECCV)*, 2020. 8

[4] Mauricio Delbracio and Guillermo Sapiro. Hand-held video deblurring via efficient fourier aggregation. *IEEE Transactions on Computational Imaging*, 1(4):270–283, 2015. 8

[5] Hongyun Gao, Xin Tao, Xiaoyong Shen, and Jiaya Jia. Dynamic scene deblurring with parameter selective sharing and nested skip connections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3848–3856, 2019. 2

[6] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7036–7045, 2019. 1

[7] Amit Goldstein and Raanan Fattal. Blur-kernel estimation from spectral irregularities. In *European Conference on Computer Vision*, pages 622–635. Springer, 2012. 1

[8] Yuanbiao Gou, Boyun Li, Zitao Liu, Songfan Yang, and Xi Peng. Clearer: Multi-scale neural architecture search for image restoration. *Advances in Neural Information Processing Systems*, 33, 2020. 1, 2, 3, 8

[9] Yong Guo, Yongsheng Luo, Zhenhao He, Jin Huang, and Jian Chen. Hierarchical neural architecture search for single image super-resolution. *IEEE Signal Processing Letters*, 27:1255–1259, 2020. 8

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[11] Shoukang Hu, Sirui Xie, Hehui Zheng, Chunxiao Liu, Jianping Shi, Xunying Liu, and Dahua Lin. Dsnas: Direct neural architecture search without parameter retraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12084–12092, 2020. 2

[12] Tae Hyun Kim, Byeongjoo Ahn, and Kyoung Mu Lee. Dynamic scene deblurring. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3160–3167, 2013. 2

[13] Tae Hyun Kim and Kyoung Mu Lee. Segmentation-free dynamic scene deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2766–2773, 2014. 2

[14] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8183–8192, 2018. 2

[15] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8878–8887, 2019. 2

[16] Zhihang Li, Teng Xi, Jiankang Deng, Gang Zhang, Shengzhao Wen, and Ran He. Gp-nas: Gaussian process based neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1

[17] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan L Yuille, and Li Fei-Fei. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 82–92, 2019. 2

[18] Hanxiao Liu, Karen Simonyan, Oriol Vinyals, Chrisantha Fernando, and Koray Kavukcuoglu. Hierarchical representations for efficient architecture search. *arXiv preprint arXiv:1711.00436*, 2017. 2

[19] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *International Conference on Learning Representations*, 2018. 3

[20] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 6

[21] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3883–3891, 2017. 1, 2, 3, 5, 6

[22] Yuesong Nan and Hui Ji. Deep learning for handling kernel/model uncertainty in image deconvolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2388–2397, 2020. 8

[23] Yuesong Nan, Yuhui Quan, and Hui Ji. Variational-em-based deep learning for noise-blind image deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3626–3635, 2020. 8

[24] Thekke Madam Nimisha, Akash Kumar Singh, and Ambasamudram N Rajagopalan. Blur-invariant deep learning for blind-deblurring. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4752–4760, 2017. 2

[25] Jinshan Pan, Zhe Hu, Zhixun Su, and Ming-Hsuan Yang. Deblurring text images via l0-regularized intensity and gradient prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2908, 2014. 1

[26] Liyuan Pan, Yuchao Dai, Miaomiao Liu, and Fatih Porikli. Simultaneous stereo video deblurring and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4382–4391, 2017. 1

[27] Hieu Pham, Melody Y Guan, Barret Zoph, Quoc V Le, and Jeff Dean. Efficient neural architecture search via parameter sharing. *arXiv preprint arXiv:1802.03268*, 2018. 2

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image com-*

puting and computer-assisted intervention, pages 234–241. Springer, 2015. 2

[29] Christian J Schuler, Michael Hirsch, Stefan Harmeling, and Bernhard Schölkopf. Learning to deblur. *IEEE transactions on pattern analysis and machine intelligence*, 38(7):1439–1451, 2015. 2

[30] Ziyi Shen, Wenguan Wang, Xiankai Lu, Jianbing Shen, Haibin Ling, Tingfa Xu, and Ling Shao. Human-aware motion deblurring. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5572–5581, 2019. 2

[31] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1279–1288, 2017. 1, 2, 5, 6, 8

[32] Masanori Suganuma, Mete Ozay, and Takayuki Okatani. Exploiting the potential of standard convolutional autoencoders for image restoration by evolutionary search. In *International Conference on Machine Learning*, pages 4771–4780. PMLR, 2018. 8

[33] Maitreya Suin, Kuldeep Purohit, and AN Rajagopalan. Spatially-attentive patch-hierarchical network for adaptive motion deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3606–3615, 2020. 2, 3

[34] Jian Sun, Wenfei Cao, Zongben Xu, and Jean Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 769–777, 2015. 2, 5, 6

[35] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8174–8182, 2018. 1, 2, 5, 6

[36] Gerard Jacques van Wyk and Anna Sergeevna Bosman. Evolutionary neural architecture search for image restoration. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019. 8

[37] Ning Wang, Yang Gao, Hao Chen, Peng Wang, Zhi Tian, Chunhua Shen, and Yanning Zhang. Nas-fcos: Fast neural architecture search for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11943–11951, 2020. 1, 2

[38] Sirui Xie, Shoukang Hu, Xinjiang Wang, Chunxiao Liu, Jianping Shi, Xunying Liu, and Dahua Lin. Understanding the wiring evolution in differentiable neural architecture search. *arXiv preprint arXiv:2009.01272*, 2020. 2

[39] Li Xu, Jimmy SJ Ren, Ce Liu, and Jiaya Jia. Deep convolutional neural network for image deconvolution. In *Advances in neural information processing systems*, pages 1790–1798, 2014. 2

[40] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *International Conference on Learning Representations*, 2016. 2

[41] Qihang Yu, Dong Yang, Holger Roth, Yutong Bai, Yixiao Zhang, Alan L. Yuille, and Daguang Xu. C2fnas: Coarse-to-fine neural architecture search for 3d medical image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[42] Yuan Yuan, Wei Su, and Dandan Ma. Efficient dynamic scene deblurring using spatially variant deconvolution network with optical flow guided training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 5, 6

[43] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *CVPR*, 2021. 2, 3

[44] Chris Zhang, Mengye Ren, and Raquel Urtasun. Graph hypernetworks for neural architecture search. *arXiv preprint arXiv:1810.05749*, 2018. 2

[45] Hongguang Zhang, Yuchao Dai, Hongdong Li, and Piotr Koniusz. Deep stacked hierarchical multi-patch network for image deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5978–5986, 2019. 2, 3, 5, 6, 7, 8

[46] Haokui Zhang, Ying Li, Hao Chen, and Chunhua Shen. Memory-efficient hierarchical neural architecture search for image denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3657–3666, 2020. 3, 8

[47] Jiawei Zhang, Jinshan Pan, Jimmy Ren, Yibing Song, Linchao Bao, Rynson WH Lau, and Ming-Hsuan Yang. Dynamic scene deblurring using spatially variant recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2521–2529, 2018. 2, 5, 6, 7

[48] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Bjorn Stenger, Wei Liu, and Hongdong Li. Deblurring by realistic blurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2737–2746, 2020. 2

[49] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual non-local attention networks for image restoration. *International Conference on Learning Representations*, 2019. 2

[50] Zhao Zhong, Junjie Yan, Wei Wu, Jing Shao, and Cheng-Lin Liu. Practical block-wise neural network architecture generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2423–2432, 2018. 2

[51] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018. 2